MODELING AND IMPLEMENATION OF ROBOTIC ARM

BY

AMIN ABDELGADER MOHAMMED FADLALLA

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

MECHANICAL ENGINEERING

MAY 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **AMIN ABDELGADER MOHAMMED FADLALLA** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN MECHANICAL ENGINEERING**.

Thesis Committee

26/05/2015
Dr. Mehmet Sunar (Adviser)

Dr. Zuhair Gasem
Department Chairman

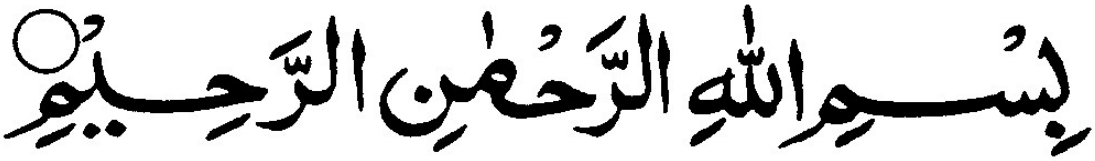
May 26, 2015
Dr. Mahir Hassan (Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Sayyid Anas Vaqar (Member)

1/6/15
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES



in the Name of Allah
the Most Compassionate
the Most Merciful

To my Father, Mother, brothers and sisters, may peace of Allah be upon them all.

ACKNOWLEDGEMENTS

First of all, I would like to thank Allah for giving me strength and health to complete this work. I am also grateful to Mechanical Engineering Department of King Fahd University of Petroleum & Minerals for giving me this opportunity to pursue my MS degree. Financial assistance received from NSTIP project # 09-ELE786-04 is gratefully acknowledged as well. Special thanks also go to my adviser Dr. M. Sunar, and to the committee members Dr. Mahir Hassan and Dr. Sayyid Anas Vaqar for their great assistance throughout this work. Last but not least, help of my close friends, namely Ahmed Tayeb, Mohanad Mohamed, Mohamed Abdeen and Siddig Mustafa, is highly valued.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Γ_{ijk}	Christoffel Symbol
$\mathcal{M}$	The Generalized Inertia Matrix
ζ	Joint Twist
g	Gravity Force's Vector
T_i^{i-1}	The Homogenous Transformation Matrix between Frame i and $i-1$
C	Coriolis Matrix
D	Inertia Matrix
DH	Denavit-Hartenberg
EE	End Effector
FBL	Feedback Linearization
J	Manipulator Jacobian
PD	Position Derivative
PE	Product of Exponentials
PID	Proportional-Integral-Derivative

THESIS ABSTRACT

NAME: Amin Abdelgader Mohammed Fadlalla

TITLE OF STUDY: Modeling and Implemenation of Robotic Arm

MAJOR FIELD: Mechanical Engineering

DATE OF DEGREE: May 2015

Robotics is a multidisciplinary subject that requires an integration of different techniques from related disciplines such as mechanics, electronics and control, that makes robotics a rich area of research, in terms of kinematics, dynamics and control. In this work, modeling and implementation of robotic arms are considered. An existing robotic manipulator having a servo at the base, at three other turning joints and at the gripper is modeled and implemented. First, inverse kinematics problem is solved and applied to the available manipulator, where different joint angles calculated via inverse kinematics are fed through Java programming. Then the dynamic model of the robot arm, that is essential for control, is derived. After that, control schemes, namely PID and feedback linearization, to restrain the motion of the manipulator are applied. Furthermore, simulation of the robotic arm by the robotics toolbox for Matlab is considered to test the motion of the system. The obtained results are discussed and analyzed. After tuning the parameters of the PID control by the differential evolution

technique, the controlled behavior of the manipulator is presented with improved overshoot, rise and settling times. The feedback linearization scheme, on the other hand, results in reasonable performance for both position and trajectory tracking controls.

MASTER OF SCIENCE

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

Dhahran 31261, Saudi Arabia

مستخلص الرسالة

الاسم أمين عبد القادر محمد فضل الله

عنوان الرسالة نمذجة وتطبيق الذراع الألي

التخصص الهندسة الميكانيكية

تاريخ التخرج مايو، 2015

علم دراسة الإنسان الآلي هو علم متعدد الجوانب، مما يجعله يتطلب تكامل المعرفة والوسائل من مجالات مترابطة كالميكانيكا والإلكترونات والتحكم. الأمر الذي يجعل علم دراسة الإنسان الآلي منطقة غنية بالبحوث من ناحية علم الحركة المجردة، علم دراسة القوى، والسيطرة. نعرض في في غضون هذه الرسالة نمذجة و تطبيق الذراع الآلي (إستنتاج العلاقات الرياضة الحاكمة وتطبيقها على الذراع الآلي المجسم). تمت نمذجة و تتطبيق ذراع آلي موجود يحتوي على محرك عند القاعدة، عند ثلاث وصلات دوارنية أخرى وعند الماسك . أولاً، مسألة علم الحركة العكسية تم حلها وتطبيقها على الذراع المتاح حيث تم إدخال الزوايا المختلفة المحسوبة في مسألة علم الحركة العكسية عن طريق البرمجة بلغة جافا. بعد ذلك اتم إستنباط العلاقات الرياضية الحاكمة، التي تعتبر أساسية لعملية السيطرة. بعد ذلك تم تطبيق إستراتيجيات تحكم\سيطرة من نوع المسيطر التناسبي-التكاملي-التفاضلي و التغذية الرجعية الخطية لتقييد حركة الذراع الآلي. علاةً على ذلك، تمت مُحَاكَاة الذراع الآلي بإستخدام صندوق الأدوات لعلم الإنسان الآلي بناءً على برنامج الماتلاب لإختبار

حركة المنطومة. تم تحليل ومناقشة النتائج المتوصلة. بعد ضبط معاملات المتحكم التناسبي-التكاملي-التفاضلي بإستخدام تقنية التطور التفاضلية، تم عرض السلوك المتحّكم بمعدل تخطي، زمن طلوع، وزمن إستقرار محسّنة. طرية التغذية الرجعية الخطية، من ناحية أخرى، نتجت في آداء معقول لكل من سيطرة الموضع النقطي وسيطرة المسار.

ماجستير العلوم

جامعة الملك فهد للبترول والمعادن

الظهران ٣١٢٦١، المملكة العربية السعودية

CHAPTER 1

INTRODUCTION

This chapter includes basic framework about the subject of robotics, literature review, objectives of this work along with the assumptions and outline of the thesis.

1.1 Background

The term robotics refers to the study and use of robots. Asimov first adopted the term in 1941 through his short science fiction story, Runaround. In the literature, more than one definition, of the robot, exists. Based on the Robotics Institute of America (RIA) definition: "A robot is a re-programmable multi-functional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks." [1]

Robot is an autonomous system that can sense its environment and act on it to achieve some goals [2]. A robot is also an automatic mechanical device often resembling a human or animal. Modern robots are usually electro-mechanical machines guided by computer programs or electronic circuitry [3]. From the engineering point of view, robots are complex, versatile devices that contain a mechanical structure, a sensory and an automatic control system. Theoretical fundamentals of robotics rely on the results of research in mechanics, electronics, automatic control, mathematics and computer sciences [1]. Classical kinematics and dynamics of robots has its roots in the work of great scientists of the past four centuries who established the methodology

and understanding of the behavior of dynamic systems. The development of dynamic science, since the beginning of the twentieth century, has moved toward the analysis of controllable man-made systems. Therefore, merging the kinematics and dynamics with the control theory is the expected development for robotic analysis. The other important development is the fast growing capability of accurate and rapid numerical calculations, along with intelligent computer programming.

1.2 Literature Survey

Nowadays, robotics is a rich area of research, in terms of their kinematics, dynamics, simulation and control. Hereafter, some of the related work to these issues are presented.

1.2.1 Kinematics

On one hand, kinematics plays a significant role in robotics and especially for the study of industrial manipulator behavior. Furthermore, a decisive part in any robotics system is the modeling and analysis of the robot kinematics [4]. It can be divided generally into forward and inverse kinematics. The former refers to direct finding of the end effector (EE) position and orientation for the given joint coordinates (joint angles for example), on the contrary, the latter, which is much more difficult, is the determination of joint variables in terms of the EE position and orientation [1]. In other words, the inverse kinematics is the process of obtaining a configuration space that corresponds to a given work space.

A number of methods and their combinations could be used to solve the inverse kinematics [5]. Several approaches have been investigated and compared by Aristidou et al. [6], however, each method beside its advantages has some limitations. Therefore, applying more than one method provides greater accuracy. Jasjit Kaur et al. [7] have analyzed and simulated a robotic arm having three links-manipulator, where after

solving the inverse kinematics resulting in two solutions out of which the best-fit solution has been selected with the help of the genetic algorithm. They have concluded that the robotic arm movement optimization can be achieved by the genetic algorithms in a practical and effective way. In [8] , 2 and 3 link-robot manipulators have been modeled and simulated; the Denavit-Hartenberg parameters have been applied to determine the coordinate transformation matrices through their different orientations. The mathematical model has also been formulated using the Newtonian-Euler modification. The simulated results have also been plotted. Thus, various graphics corresponding to the above analysis have been plotted. A geometric approach to solve the unspecified joint angles needed for the autonomous positioning of a robot arm is presented in [4]. Based on few assumptions concerning the working environment of the robot and the kind of manipulation required, beside the usage of basic trigonometric, an easier solution compared to others has been proposed. The introduced methodology has been tested using five degrees of freedom robotic arm compounded to an i-Robot create platform. By carrying out the mechanical design, all components have been assembled together in addition to two infrared sensors to detect the object and find its position. Finally, the electrical design was done, and the proposed method was tested and justified. However, this approach lacks the generality due to the abundant assumptions considered, and what is more is that the last link is restricted to just two orientations. In what is suggested here, the last link could have any arbitrary orientation, which will make the analysis more general to different configurations.

1.2.2 Dynamics

One other hand, obtaining the correct dynamic model of a robot is a necessary step for the control [1] . In other words, to achieve a proper control, a valid dynamic model must be in hand. In their approach of modeling and identification of high-performance robot control, Kostic et al. [9], highlighted a procedure for obtaining kinematics and

dynamics models of a robot that are apt for robot control design. That involves: deduction of robot kinematics and dynamics models and building their correctness, estimation of the model parameters experimentally, validation of the model, and identification of the abiding (remaining) robot dynamics which should not be ignored if robustness and high-performance robot operation were required. Frictional and inertial parameters were also estimated. While kinematics relates the joint or configuration space to the task/work space, dynamics on other hand connect the motions, speeds, and accelerations with the applied forces or torques (control input). Although the forward kinematics/dynamics is straightforward, unfortunately the derivation of compact form models for inverse kinematics/dynamics is not an easy task. The derivation requires a series of processes accompanied by undying simplifications of intermediate results. Finally, a straightforward but efficient estimation of parameters of the rigid-body dynamic model that includes friction effects is suggested. With these additional dynamics available, more advanced feedback control designs become possible. Modeling and implementation of a mobile robotic arm for industrial tasks was introduced by Ahad et al. [10], where the technique of interfacing the robotic arm and gear/stepper motors with the programmed micro-controller used to govern the operation of the robot was presented. They based their method in designing and constructing the robotic arm on the operational features and properties of the micro-controllers, gear motors, stepper motors and their programming, in addition to the electronic circuit diagram. The mechanical components like the base, motors, and other parts were arranged and built in a systematic way that produced a balanced and firm structure. Flowcharts of the operation of the robot and control circuit diagram were presented, where the micro-controller sends signals to different motors which move the mobile robot accordingly. It is found that the developed model minimizes the complexity of wrist and gripper compared to earlier methods. Mechanical parts could also be easily replaced because the mobile platform, gripper, wrist and elbow were not enduringly

attached with one another. In [11], Long et al. showed model and simulation of the articulated robotic arm test system for combination drive. Their contributions is to create a test system of a new smaller volume, articulated robotic arm, to develop the mathematical model of the system, and to execute the numerical simulation and PID controller optimization of the system in Matlab environment. Usually, the driving force of the robotic arm supplied by a motor (stepper, servo etc.) is not too large and it could be used to manipulate light loads in general. When the load need to be manipulated is large, the required power from the servo is high, necessitating the usage of bigger servo motor to move the arm, which will definitely increase the volume of the robot manipulator and therefore, its design will be more difficult. Thus the construction of smaller volume, strong robot capable of moving great loads is a challenging task. So, the proposed methodology was to combine servo motor and hydraulic motor in the driving system. The servo system is basically at the helm of active exercise of the robot, while the hydraulic motor is liable for auxiliary afterburning during the dynamic reaction of the articulated robotic arm. The high driving power of the hydraulic system compensates for the limitation of the small carrying capacity of the servo, and therefore, improves the output force of the driving system of the robot. Mathematical model of the robot has allowed for the Matlab numerical simulation and PID optimization. According to the synchronized control and PID optimization results, this system combines the advantages of strong driving capability for the hydraulic drive system together with the high positioning accuracy for the motor drive system. Beside their uses in industry [12–14] robotics has applications in biologic and medical fields, where it has been used to help disabled people to perform the necessarily daily living activities. Methods for kinematics modeling of robotics and biological systems are given in [15] . The purpose was to construct strategies that automatically produce kinematics models for the movements of biological and robotic systems. Particular attention was paid to build a wheelchair-mounted robot

arm integrated to be used by children with quadriplegia. The movement of the system is approximated with an open kinematic chain. Two identification methods are adopted to generate kinematics models automatically and tested with planar (RR) robotic arm; and close results to the actual parameters are achieved. While the first method needs the elimination of the displacement variables that could be measured, the second method strives to estimate the changes in these variables.

1.2.3 Simulation

Because robotics is a multidisciplinary subject, the existence of an instrument that somehow combines the related disciplines in one platform will be indispensable in robotics. Fortunately, simulation has been recognized as a powerful tool of planning, visualization, and strategic technique in different areas of research and development. Simulation also plays a very important role in robotics [16] . Active learning in robotics based on simulation tools was introduced in [17] . A simulation methodology has been described together with the new simulation tools, RobotScene and SGRobot. It is argued that the activity has been carried out successfully for several years; and the experience is such that it motivates the students and improves their awareness of the theoretical concepts involved. Thus, benefits of active learning have been admitted and, in the framework of new methodologies, laboratory, and project oriented courses are encouraged. Furthermore, the success of performing those activities is guaranteed, provided that prior knowledge form the students is completed. In [18] , development of a PC-based simulation and control platform for a 6-DOF robotic arm is given. The system basically includes software platform and servo control card based on a microcontroller, designed to move the servo motor connected to each joint of the robot. The developed software was able to draw and simulate the 3D kinematic pattern of the robot. With the help of the GUI simulation, the user could visualize the manipulator motion planning. Besides, the user also could control the real robotic arm through

that software. Lastly, point-to-point and continuous path motions were all tested in simulation and real robot control.

Ranjan et al. [19] , proposed modeling and simulation of robotic humanoid arm, where the kinematics equations were derived based on Denavit-Hartenberg (DH) notation. Subsequently, the inverse kinematics parameters for the motion trajectory were obtained. A humanoid robot is formed of replica of parts which function in a symmetric fashion like a human. The principal three functions are walking legs, working hands and stereoscopic viewing eyes. Although many stable walking machines have been developed in this fashion, but they do not fulfill expectations of a robot that could give impression of a human-like movement. However, the use of considerable number of actuators, sensors and other parts has made it a very costly affair. Low cost realizations in the field still need more investigation. For Matlab simulation purposes, it is assumed that the position of the object is kept in the workspace; given the dimensions of the arm and elbow, a two link-block model has been generated in Simulink; and therefore, path planning of the robot's EE has been specified. Geometrical model of the robotic humanoid arm has been developed utilizing forward and inverse kinematics equations through the DH convention. Simulated virtual motion of the robotic arm has been observed, which represents the first step in actually controlling the physical system. With the simulation, control signals to be used for the mechanical control of the robot can be achieved. The control law for achieving the desired speed and precision could also be implemented in the simulation [19].

1.2.4 Control

After modeling and simulation of the robotics system, a control strategy is applied to govern the motion of the system. The control of robot manipulators is unlike that of other industrial equipments, because, it is companied with a big number of separately controllable mechanical axes [20] . Robust control of a two-link rigid manipulator

was introduced by Yadav and Singh in [21] , where the Lagrange-Euler method was used to drive the dynamics of the arm and the linear fractional transformation for the uncertainty in the model (such as moment of inertia, friction and actuators). Two different robust control schemes, H_∞ and μ-synthesis were designed using Matlab to control the robotic manipulator. Although both controllers were proficient of stabilizing the arm in a very effective manner, the μ-synthesis controller has been recognized to have superior robust performance. In [22], fuzzy control for the under-actuated manipulator is given. The complex mathematics usually encountered with the nonlinear dynamic model of the manipulator could be avoided making the proposed method very simple and computationally fast. For verification purposes, numerical simulation for the position control of a planar (RRR) was presented. From the obtained results, the introduced method is shown to provide an effective way for the intelligent control of second-order nonholonomic systems. Position control of a single-link robot arm using a multi-loop PI controller is presented in [23]. In addition, the computed torque control (CTC) of the PUMA 560 robot manipulator was investigated by Piltan et al. [24], based on the Matlab and Simulink realization. PD-CTC and PID-CTC have been tested to step and ramp responses, where simulation results show a good performance of the proposed methodology. Similar approach using the sliding mode control (SMC) is followed in [25]. A Neural network (NN) self tuning PID controller is proposed in [26]. It is shown with simulation results that introducing neural network with the conventional PID control improves the performance and reduces tracking errors significantly. Simulation curves further show that, initially the performance is not accurate but after one second it becomes stable, due to the fact that the NN has not learned yet in the beginning. Danut Receanu [27] introduced modeling and simulation of the nonlinear computed torque control in Simulink/MATLAB for an industrial robot, that can fully compensate the nonlinear Coriolis and centripetal forces. A PD control scheme is used to control the motion of a two-link planar robot, whose

dynamic model is based on the Euler-Lagrange equation. It is concluded that when the friction is compensated, the steady-state error roughly disappears, but the settling time increases.

1.2.5 Practical Implementation

In [28], Menon et al. Introduced motion planning for Smooth grasping of moving objects. They proposed a search-based kinodynamic motion planning strategy that produces a time parameterized trajectory for both the manipulator and the end effector, capable of heedfully picking up the object at betimes feasible point in its trajectory. For high-dimensionality manipulation of the time-parameterized kinodynamic planning problem, the prescribed methodology utilized adaptive dynamic motion and informative heuristics primitives. The proposed methodology is tested on a 4 DOF manipulator to pick up objects out of a moving belt.

Wenzhe Wang1 et al. [18] proposed simulation and control platform for 6 DOF robot manipulator based on a PC. The developed system mostly comprises servo control card based on microcontroller STC12C5A60S2 developed to drive the servomotor connected at each joint of the manipulator and software platform. By using the Open Graphics Library (OpenGL) functions, the software is capable of drawing and simulating 3D kinematic models of the manipulator, it also introduces 3D motion planning simulation property. The user can visualize the manipulator motion planning utilizing the simulation in the GUI. Moreover, through this software, the user also can control the real robotic arm. Newton-Raphson iteration algorithm is adopted to solve inverse kinematics in the software. Both point-to-point motion and continuous path movement are all tested in simulation and real robot control. The entire system has been successfully implemented.

1.3 Objectives

The main objectives of this work can be summarized as follows:

i . To develop an inverse kinematics model of a physically existing robotic arm and to verify the model with the real-time experiment.

ii . To determine the dynamic model of the robot.

iii . To construct two different control strategies for motion control of the system.

iv . To simulate the robotic arm at different configurations.

1.4 Assumptions

While modeling and simulating of the robot arm, following assumptions are made:

- Frictional forces at the manipulator joints are small and hence can be neglected,
- Actuator dynamics are ignored, but commands can be sent to joints to exert torques on them,
- The motion of the gripper (end-effector) does not play a significant role in dynamics, excluding that making our system a 4 DOF articulated robot, and
- The task space is free form obstacles.

1.5 Thesis Outlines

The rest of this document is organized as follows: In chapter II, the kinematics model of the robot is considered based on the Denavit-Hartenberg convention and product of exponentials formula . The inverse kinematics problem is then discussed and the Jacobian of the robot is presented. Moreover, the simulation of the robot is introduced

based on the Robotic Toolbox for Matlab, The chapter is concluded by presenting some experimental work has been performed while implementing the inverse kinematics solution. The dynamics model of the robot is developed in chapter III utilizing the concept of the Jacobian introduced in the previous chapter . Prior to this development, dynamics of two and three link-planar robots are studied to pave way for the 4 DOF robotic arm under investigation. Chapter IV makes use of the dynamic model of the robot to implement two control strategies, namely Proportional-Integral-Derivative (PID) control and Feedback Linearization (FBL) control to control the motion of the robot. Results of feedback linearization and PID control are presented and discussed. Chapter V concludes this work and highlights several points for further investigation.

CHAPTER 2

MANIPULATOR KINEMATICS

The robot manipulator kinematics aims to find the relationship that relates the motion of the manipulator's joints and the emanating movement of the links (the rigid bodies) that construct the manipulator. Two important problems arise when we deal with the kinematic analysis of robots: the direct kinematics problem and the inverse kinematics problem. This chapter first presents an introduction showing the purpose of the kinematics modeling. Then the forward and inverse kinematics problems are discussed. After that velocity analysis is presented and the Jacobian of the robot manipulator is investigated. In addition to some simulation results obtained by simulating the motion of the robot, some experimental procedures implementing the outcomes of the inverse kinematics solution along with a brief description of the available setup kits are provided.

2.1 Introduction

The major objective is to control both the position and orientation of the manipulator's end effector (gripper) in its work space. The information is sent to all the joint servos in order to move the gripper of the robotic arm to the desired position in the workspace. In order to program the robotic arm motion, we first derive the relationship that relates the position and orientation of the gripper to the joint variables,

which is called the manipulator's kinematic model. The robotic arm under study is shown in figure (2.1).

Figure 2.1: 5 DOF Robotic Arm

Various rotary and linear motions of the robotic arm are shown in Figure 2.2 . Each motion is made possible by a servomotor (or just servo) at the corresponding location.

2.2 Kinematic Model

The kinematics model of the robot manipulator under study is derived using two approaches, the first method is so called the Denavit-Hartenberg convention [1] which is commonly used in robotic literature, and the other one is the product of exponentials method [29, 30] . And we'll show that both techniques lead to the same kinematics model, which ensure the correctness of the model. This is important, because the dynamic model derived later is built on this kinematics model.

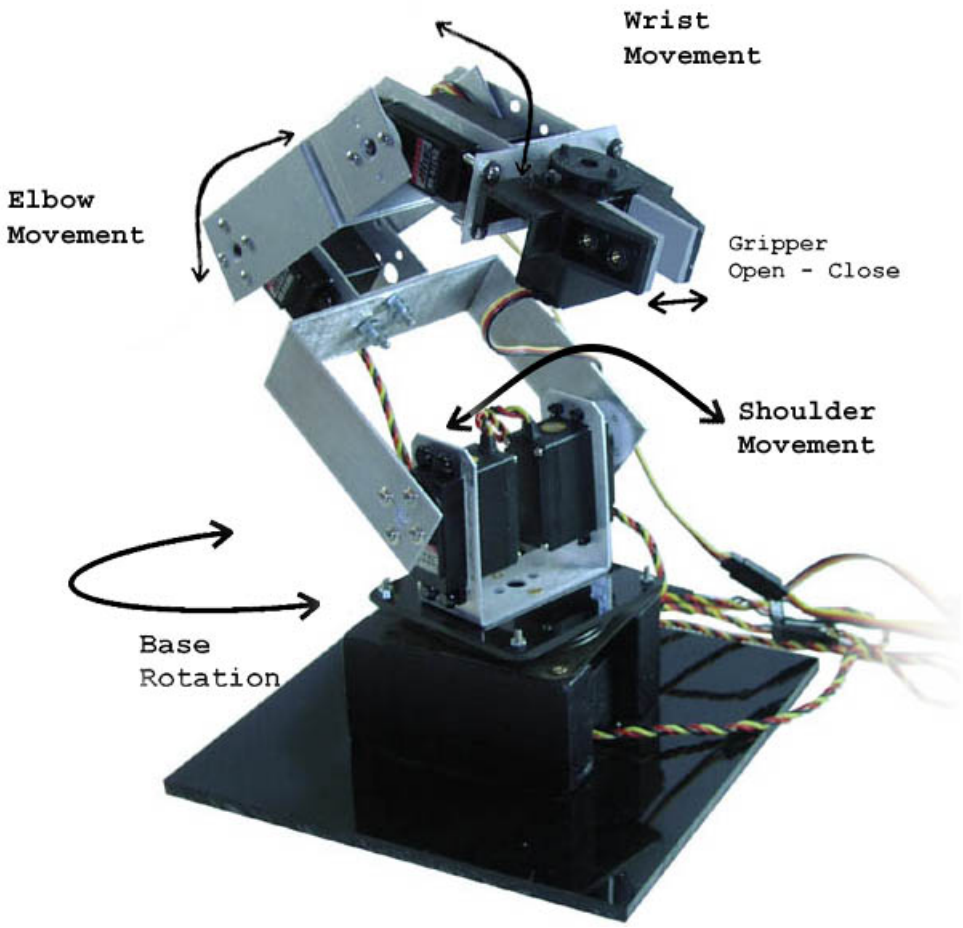


Figure 2.2: Robotic arm with five independent motions

2.2.1 Denavit-Hartenberg convention

The DH is commonly used in the kinematics analysis of the robotic manipulator. It is based on attaching a coordinate frame at each joint and specifying four parameters known as DH parameters for each link, and further utilizing these parameters to construct the DH table. Finally, a transformation matrix between different coordinate frames is obtained. As stated before, the major objective is to control both the position and orientation of the end effector (EE) or the gripper in its work space. We will first derive the relationship between the joint variables, the position and orientation of the

gripper, using the DH method. The robotic arm with DH parameters is presented in figure (2.3).

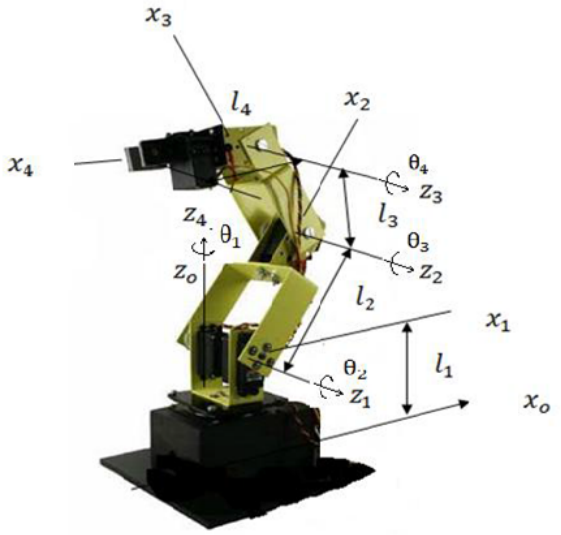


Figure 2.3: DH parameters

Forward Kinematics

As Figure (2.3) depicts a coordinate frame has been attached to every link in order to determine its configuration in the neighboring frames using the rigid body motion method. To do so a DH table needs to be constructed as follows. By applying the

Table 2.1: DH parameters

Frame # i	a_i	α_i	d_i	θ_i
1	0	90	l_1	θ_1
2	l_2	0	0	θ_2
3	l_3	0	0	θ_3
4	l_4	0	0	θ_4

DH notations for the joints coordinates, the DH-table can be constructed as listed in table (2.1). The link lengths as shown in Figure (2.3) are: $l_1 = 11.5\text{cm}, l_2 =$

12cm, and $l_3 = l_4 = 9$cm.

Forward Transformation Matrices

Once the DH table is built, the transformation matrices can be found. Generally, the transformation matrix from the frame B_i to the frame B_{i-1} for the standard DH method is given by [1]:

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Then the individual transformation matrices for $i = 1, 2, 3,$ and4 can be easily obtained.

$$T_1^0 = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & \mathrm{L}_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_2^1 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & \mathrm{L}_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & \mathrm{L}_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_3^2 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & \mathrm{L}_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & \mathrm{L}_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_4^3 = \begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & \mathrm{L}_4\cos(\theta_4) \\ \sin(\theta_4) & \cos(\theta_4) & 0 & \mathrm{L}_4\sin(\theta_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Thereafter, the complete transformation T_4^0 can be found from:

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 \tag{2.2}$$

Upon the determination of T_4^0 we can find the global coordinates of the end effector. Recall figure (2.3) the tip point of the arm is at the origin of frame B_4 i.e. it is at $[0 \quad 0 \quad 0 \quad 1]^T$ in the last frame. So its position in the global frame (the base frame) becomes:

$$r_p^0 = T_4^0 r_p^4 = T_4^0 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} r_14 \\ r_{24} \\ r_{34} \\ 1 \end{bmatrix} \tag{2.3}$$

which is the last column of the transformation matrix T_4^0.

After simplifications using trigonometric formulas, previous equations become:

$$x = r_{14} = \cos\theta_1(l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3) + l_4\cos(\theta_2+\theta_3+\theta_4)) \tag{2.4}$$

$$y = r_{24} = \sin\theta_1(l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3) + l_4\cos(\theta_2+\theta_3+\theta_4)) \tag{2.5}$$

$$z = r_{34} = l_1 + l_2\sin\theta_2 + l_3\sin(\theta_2+\theta_3) + l_4\sin(\theta_2+\theta_3+\theta_4) \tag{2.6}$$

where x , y , and z are the global coordinates of the end effector. In addition, the end effector orientation is:

$$\phi = \theta_2 + \theta_3 + \theta_4 \tag{2.7}$$

Nevertheless, DH parameters are akin to the configuration of the robot. For different manipulator structures, the kinematics equations are not unique. Moreover, kinematics equations of the manipulator based on the DH convention provide some singularity, making the equations difficult to solve or unsolvable in many cases. Also, in DH convention, the common normal is not properly specified when the two joint axes are parallel. In such situation, the DH convention possess a singularity, since with a small deviation in parallel joint axes spatial positions can produce a great change in the DH coordinate representation of their corresponding location. In the following section, an alternative to the DH convention is presented.

2.2.2 Product of Exponentials Formula

Aside from the DH convention, the other method is the so-called product of exponentials formula, which will be presented in details in this section. This method represents the kinematics of an open-chain mechanism as the product of exponentials of twists. This setting works whenever the joints of the robot consist of either revolute, prismatic or helical joints, which is practically the case for all commercially available robot manipulators. It provides a global, geometric representation of the kinematics of a manipulator, which greatly simplifies the analysis of the mechanism and provides a very structured parameterization for open- chain robots. Using the reality that the motion of each joint is produced by a twist accompanied with the joint axis, a more geometric representation of the kinematics can be acquired. Taking ζ as a twist, the forward kinematics is given by [30] :

$$g_{st}(\theta) = e^{\hat{\zeta}_1\theta_1}e^{\hat{\zeta}_2\theta_2}...e^{\hat{\zeta}_n\theta_n}g_{st}(0) \tag{2.8}$$

Equation (2.8) is denoted the product of exponential formula for the direct kinematics of the robot . $g_{st}(0)$ is the initial configuration, $g_{st}(\theta)$ is the final configuration

of the robot and $e^{\hat{\zeta}_n \theta_n}$ is a matrix exponential given by [31]:

$$e^{\hat{\zeta}_n \theta_n} = \begin{bmatrix} e^{\theta_n \hat{\omega}_n} & (I - e^{\theta_n \hat{\omega}_n})(\omega_n \times v_n) + \theta_n \omega_n \omega_n^T \theta_n \\ 0 & 1 \end{bmatrix} \tag{2.9}$$

For a revolute joint (as in our case) the twist ζ_i is given by:

$$\zeta_i = \begin{bmatrix} -\omega_i \times q_i \\ \omega_i \end{bmatrix}$$

where $\omega_i \in R^3$ is a unit vector in the twist axis direction and $q_i \in R^3$ is any point on the axis. In case if we've a prismatic joint,

$$\zeta_i = \begin{bmatrix} v_i \\ 0 \end{bmatrix}$$

where $v_i \in R^3$ is a unit vector directing in the translational direction.

$$\zeta_1 = \begin{bmatrix} -\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \zeta_2 = \begin{bmatrix} -\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ l_1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} l_1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\zeta_3 = \begin{bmatrix} -\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \times \begin{pmatrix} l_2 \\ 0 \\ l_1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} l_1 \\ 0 \\ -l_2 \\ 0 \\ -1 \\ 0 \end{bmatrix} \qquad \zeta_4 = \begin{bmatrix} -\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \times \begin{pmatrix} l_2 + l_3 \\ 0 \\ l_1 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} l_1 \\ 0 \\ -(l_2 + l_3) \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

The initial configuration of the robot can be expressed as

$$g_{st}(\theta) = \begin{bmatrix} & \begin{bmatrix} l_2 + l_3 + l_4 \\ 0 \\ l_1 \end{bmatrix} \\ I & \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

The robot direct kinematics map has the form

$$g_{st}(\theta) = e^{\hat{\zeta}_1 \theta_1} e^{\hat{\zeta}_2 \theta_2} e^{\hat{\zeta}_3 \theta_3} e^{\hat{\zeta}_4 \theta_4} g_{st}(0) \tag{2.10}$$

By spreading the components in the formula of exponentials product we get:

$$g_{st}(\theta) = \begin{bmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix}$$

where

$$R(\theta) = \begin{bmatrix} \cos(\theta_2+\theta_3+\theta_4)\cos\theta_1 & -\sin\theta_1 & -sin(\theta_2+\theta_3+\theta_4)\cos\theta_1 \\ \cos(\theta_2+\theta_3+\theta_4)\sin\theta_1 & \cos\theta_1 & -sin(\theta_2+\theta_3+\theta_4)\sin\theta_1 \\ \sin(\theta_2+\theta_3+\theta_4) & 0 & \cos(\theta_2+\theta_3+\theta_4) \end{bmatrix} \tag{2.11}$$

$$p(\theta) = \begin{bmatrix} \cos\theta_1(l_3\cos(\theta_2+\theta_3)+l_2\cos\theta_2+l_4\cos(\theta_2+\theta_3+\theta_4) \\ \sin\theta_1(l_3\cos(\theta_2+\theta_3)+l_2\cos\theta_2+l_4\cos(\theta_2+\theta_3+\theta_4) \\ l_1+l_3\sin(\theta_2+\theta_3)+l_2\sin\theta_2+l_4\sin(\theta_2+\theta_3+\theta_4) \end{bmatrix} \tag{2.12}$$

It is clear that the above equation 2.12 is identical to those in equations 2.4, 2.5 and 2.6. Thus, the kinematics modeling using the DH convention and product of exponentials are equivalent in this case.

2.3 Inverse Kinematics

Consequently, we end up with a set of four nonlinear equations 2.4, 2.5, 2.6 and 2.7 with four unknowns. Solving these equations algebraically (or by other means), known as the inverse kinematics, requires that we need to know the joint variables θ_1, θ_2, θ_3, and θ_4 for a given EE position $[x, y, z]$ and orientation ϕ . Squaring and adding the equations after some mathematical manipulation, using trigonometric functions, we

get from equations 2.4, 2.5 and 2.6:

$$\theta_1 = \arctan\left(\frac{y}{x}\right) \tag{2.13}$$

$$\theta_2 = \arctan\left(\frac{c}{\pm\sqrt{r^2 - c^2}}\right) - \arctan\left(\frac{a}{b}\right) \tag{2.14}$$

$$\theta_3 = \arccos\left(\frac{(x - l_4 c_1 c\phi)^2 + (y - l_4 s_1 c\phi)^2 + (z - l_1 - l_4 s\phi)^2 - l_2^2 - l_3^2}{2 l_2 l_3}\right) \tag{2.15}$$

where $a = l_3 \sin\theta_3$, $b = l_2 + l_3 \cos\theta_3$, $c = z - l_1 - l_4 \sin\phi$, and $r = \sqrt{a^2 + b_2}$. Having determined θ_1 , θ_2 , and θ_3, we can then find θ_4 from the EE orientation ϕ as follows:

$$\theta_4 = \phi - \theta_2 - \theta_3 \tag{2.16}$$

This shows the algebraic solution to the manipulator inverse kinematics problem.

2.4 Velocity Kinematics

In the preceding sections, we analyzed the forward and inverse position equations that relate the joint positions to end-effector posture. Hereby we construct the corresponding relation for the velocity, relating the end-effector angular and linear velocities to velocities of the manipulator joints. Form a mathematical point of view, the equations of the manipulator kinematics introduces a function that relates the Cartesian space - of positions and orientations of the robot - to the joints space (configuration space). Then the Jacobian of this function gives the velocity relationships correspondingly. The Jacobian is a matrix valued function and can be thought of as the vector version of the ordinary derivative of a scalar function [32]. This Jacobian (the Jacobian matrix) is one of the most important quantities in the analysis and control of robot motion.

It emerges in nearly every aspect of robotic manipulation: in the problem of path planning, in singular configurations determination, in the arranged anthropomorphic motion execution, in the dynamic model derivation and when transforming forces and torques acting at the end-effector to the joints of the robot.

2.5 Jacobian

We start our discussion by considering an n-link manipulator whose joint variables defined as $\theta_1, \theta_2, ..., \theta_n$. Furthermore suppose

$$T_n^0(\theta) = \begin{bmatrix} R_n^0(\theta) & p_n^0(\theta) \\ 0 & 1 \end{bmatrix} \tag{2.17}$$

denotes the homogeneous transformation from the end effector frame to the base frame, where $\theta = [\theta_1\theta_2...\theta_n]^T$ is the vector of joint variables. The position of the end effector p_n^0 and its orientation $R_n^0(\theta)$ are both function of time when the robot moves. Thus, here we seek a relationship that relates the joint angular velocities to the end effector velocity.

Let $\tilde{\omega}_n^0 = \dot{R}_n^0(R_n^0)^T$ define the skew symmetric matrix corresponding to the vector of the angular velocity ω_n^0 of the end effector, and let $v_0^n = \dot{p}_n^0$ represent the end effector linear velocity. Our objective is to get an expressions of the form $v_n^0 = J_v\dot{\theta}$ and $\omega_n^0 = J_\omega\dot{\theta}$ where both J_ω and J_v are matrices of $3\times n$. Or, in general we looking for such a relation

$$\begin{bmatrix} v_n^0 \ \omega_n^0 \end{bmatrix} = J_n^0\dot{\theta} \tag{2.18}$$

where J_n^0 is given by the $J_v^0 = \begin{bmatrix} J_v \ J_\omega \end{bmatrix}$ The matrix J_n^0, a $6\times n$ matrix where n represents the number of links, is called the *Jacobian matrix* for the robotic manipulator or the manipulator Jacobian or simply the Jacobian. Generally, the top part of the Jacobian

(J_v) is calculated by [32]

$$J_v = [J_{v_1}.....J_{v_n}]$$

where the ith column J_{v_i} is given by

$$J_{v_i} = z_{i-1}\times(o_n - o_{i-1})$$

if joint (i) is revolute, and

$$J_{v_i} = z_{i-1}$$

if joint (i) is prismatic.

In addition, z_{i-1} is given as: $z_{i-1} = R^0_{i-1}$ where R^0_{i-1} is the rotation matrix between $(i-1)$)and the base frame.

The part at the bottom of the Jacobian expression is obtained by

$$J_\omega = [J_{\omega_1}.....J_{\omega_n}]$$

where the ith column J_{ω_i} is $J_{\omega_i} = z_{i-1}$ for the revolute joint and $J_{\omega_i} = 0$ for the prismatic joint. Thus, combining the two halves of the Jacobian together, we end up with this expression for the Jacobian of an n-link manipulator:

$$J = [J_1 J_2 ... J_n] \tag{2.19}$$

where the ith column J_i is given by

$$\begin{bmatrix} z_{i-1}\times(o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix}$$

if the joint is revolute, and

$$\begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix}$$

if the joint is prismatic.

2.6 The Jacobian of the 4-DOF manipulator

The complete Jacobian for this manipulator under study takes the form

$$J = \begin{bmatrix} z_0 \times (o_4 - o_1) & z_1 \times (o_4 - o_1) & z_2 \times (o_4 - o_2) & z_3 \times (o_4 - o_3) \\ z_0 & z_1 & Z_2 & z_3 \end{bmatrix}$$

Using the forward transformation matrices, given below, we can formulate the Jacobian for this manipulator:

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & l_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_4^3 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & l_4\cos\theta_4 \\ \sin\theta_4 & \cos\theta_4 & 0 & l_4\sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Having the four transformation matrices, the vector z_m is given by $z_m = R_m^0 k$ where R_m^0 is the rotational part of $T_m^0 = T_1^0 ....T_m^{m-1}$. The vector o_m is obtained from the first three elements of T_m^0 last column, with $o_0 = (0, 0, 0)^T$ Therefore $J = [J_1 J_2 ....J_n]$ where

$$J_1 = \begin{bmatrix} -s_1(l_3c_{23} + l_2c_2 + l_4c_{234}) \\ c_1(l_3c_{23} + l_2c_2 + l_4c_234) \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad J_2 = \begin{bmatrix} -c_1(l_3s_{23} + l_2s_2 + l_4s_{234}) \\ -s_1(l_3s_{23} + l_2s_2 + l_4s_{234}) \\ l_3c_{23} + l_2c_2 + l_4c_{234} \\ s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} -c_1(l_3s_{23} + l_4s_{234}) \\ -s_1(l_3s_{23} + l_4s_{234}) \\ l_3c_{23} + l_4c_{234} \\ s_1 \\ -c_1 \\ 0 \end{bmatrix} \qquad J_4 = \begin{bmatrix} -c_1l_4s234 \\ -s_1l_4s_{234} \\ l_4c_{234} \\ s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

combining:

$$J = \begin{bmatrix} -s_1(l_3c_{23} + l_2c_2 + l_4c_{234}) & -c_1(l_3s_{23} + l_2s_2 + l_4s_{234}) & -c_1(l_3s_{23} + l_4s_{234}) & -c_1l_4s_{234} \\ c_1(l_3c_{23} + l_2c_2 + l_4c_234) & -s_1(l_3s_{23} + l_2s_2 + l_4s_{234}) & -s_1(l_3s_{23} + l_4s_{234}) & -s_1l_4s_{234} \\ 0 & l_3c_{23} + l_2c_2 + l_4c_{234} & l_3c_{23} + l_4c_{234} & l_4c_{234} \\ 0 & s_1 & s_1 & s_1 \\ 0 & -c_1 & -c_1 & -c_1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.20}$$

2.7 Simulation of the Robotic Arm

Since robotics is a multidisciplinary subject, its study requires skills form different fields of knowledge. Thus, simulation has been recognized as a suitable tool that combines all these features together enabling the user of a direct visualization of different kinds of motion that a robot may perform, making the role of simulation very important in robotics [16]. Using the robotics toolbox for Matlab [33–35] , the kinematics of a robotic arm can be simulated and analyzed based on the DH convention described before. The toolbox takes a conventional approach to represent the kinematics and dynamics of serial-link robotic arms. Besides, it provides several functions and routines, which are handy for the simulation and scrutiny of robotic manipulators, like kinematics, dynamics and trajectory creation. Based on the aforementioned toolbox, many configurations of the robot manipulator are easily visualized. In order to simulate the robotic manipulator, first we make a vector of *link* objects and insert the four groups of DH parameters as in Table 1. Then, these are passed to the constructor *SerialLink*, which is the key step to utilize the toolbox. A detailed procedure can be found in [35]. In order to generate the plot, four joint variables are needed. In the first case, all joint variables are entered in the form of zeros (row vector), such as $[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4] = [0 \quad 0 \quad 0 \quad 0]$. Figure (4) shows the rest or the home position of the robot, where all joints variables are zeros. In the toolbox, each revolute joint is resembled by a small cylinder. Since we have four revolute joints, four cylinders can be seen in figures (2.4-2.7). In figure (2.5), the vector of the joint variables provided is $\theta = [0 \quad \pi/2 \quad 0 \quad 0]$, while in figure (2.6) the joint vector used is $\theta = [-\pi/2 \quad 0 \quad -\pi/2 \quad 0]$ and $\theta = [\pi \quad \pi/4 \quad -\pi/2 \quad \pi/4]$ for figure (2.7).

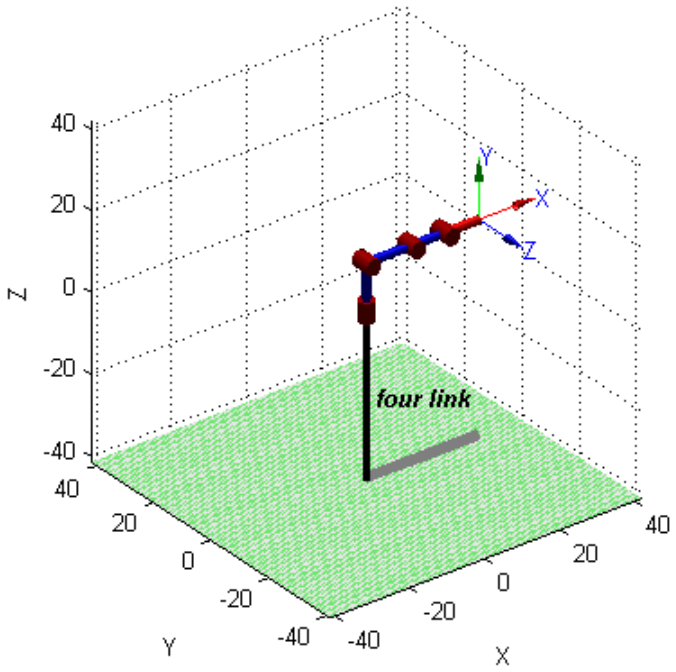


Figure 2.4: Home position

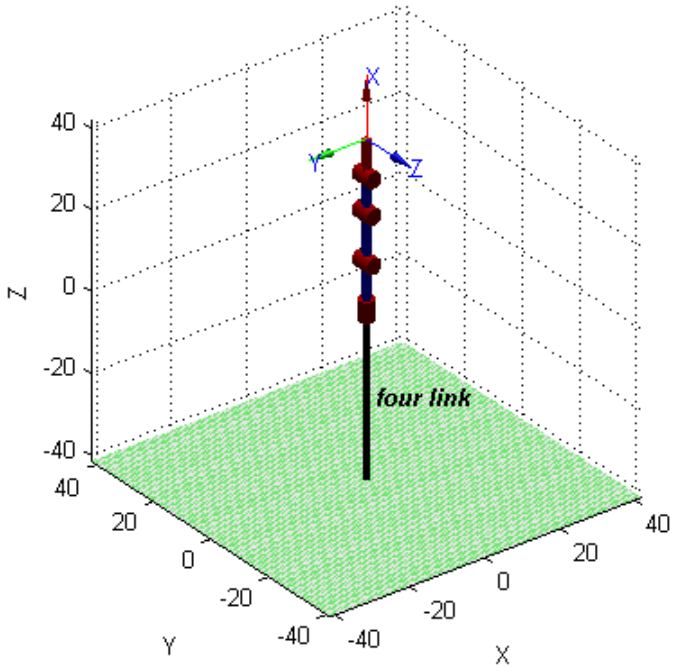


Figure 2.5: Upright position

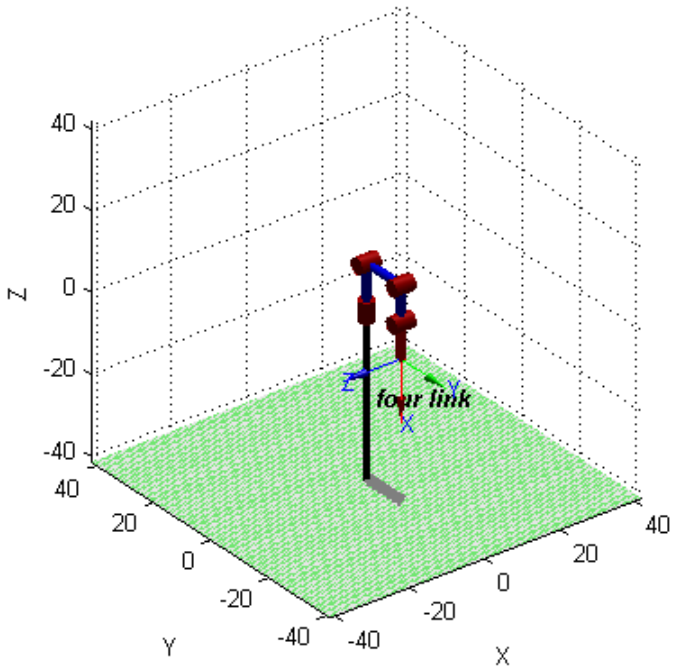


Figure 2.6: Left-down position

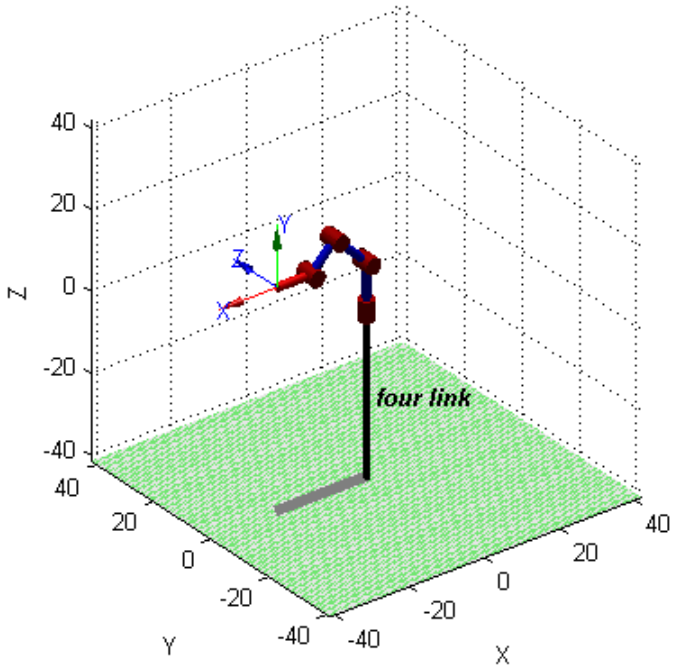


Figure 2.7: All joints are given angles

2.8 Experimental Work

This section explains the experimental work related to the inverse kinematics implementation. Java programming Language is used to communicate between the GUI of the PC and the FPGA board that in turn converts the Java commands to corresponding signals and send those to the servos, which will move the robot.

2.8.1 Java Programming for Robotic Motion

After the determination of the joint variable required to put the gripper of the robot at desired position from the inverse kinematic solution shown before (2.3), the programming stage involves the following:

- The calculated angles are converted to Java commands where each angle is resembled in two instructions.
- There is a total of 4 turning joints, for which J1, J2, J3 and J4 represent joint angular locations as 8 bit numbers. These joint locations are sent as hexadecimal numbers.
- In an interactive manner, these numbers are executed in Java language on a PC.
- In the Java program, there is an option to specify time delays between the joint motions.

A sample of the code is shown below (full program is in appendix(5.2)).

2.8.2 Testing of the System

The robotic system is composed of a dual robotic arm with a base and an FPGA-based controller. Each robotic arm has a servo at the base, at three other turning joints and at the gripper. The constructed system operates each robotic arm in a symmetric

```
// 4D 01 FF 03 D0 00 FF 00 FF 0D
this.out.write((char) 0x4d);
this.out.write((char) 0x01);
this.out.write((char) 0xff);
this.out.write((char) 0x03);
this.out.write((char) 0xd0);
this.out.write((char) 0x00);
this.out.write((char) 0xff);
this.out.write((char) 0x00);
this.out.write((char) 0xff);
this.out.write((char) 0x0d);

try {
        Thread.sleep(2000);
} catch(InterruptedException ex) {
        Thread.currentThread().interrupt();
}
```

Figure 2.8: A punch of Java code 2

fashion, similar to a person using his/her two arms to grab and lift an object. The two robotic arms used in this work, as figure 2.9 depicts, are RA-02 robotic arms of Images SI Inc [36]. The robotic arm controller system is centered around an FPGA with several hardware blocks running in parallel as shown below.

2.8.3 Servo System

Essentially, servomotors are geared dc motors contain a positional feedback control that permits the rotor to be located accurately. According to the specifications ([37]) the shaft can be located through a minimum of 90° (±45°), which can be extend closer to 180° (±90°) by modifying the positional control signal. There are three wire leads to a servomotor. Two of them are for power +5v and ground. The third one provides the motor with a position control signal. This signal is a single variable width pulse. The pulse can be ranged from 1 to 2 ms. The width of the pulse governs the position of the shaft attached to the specific servomotor. A 1-ms pulse moves the shaft to the extreme counterclockwise (CCW) location (-45°). A 1.5-ms pulse locates the shaft in

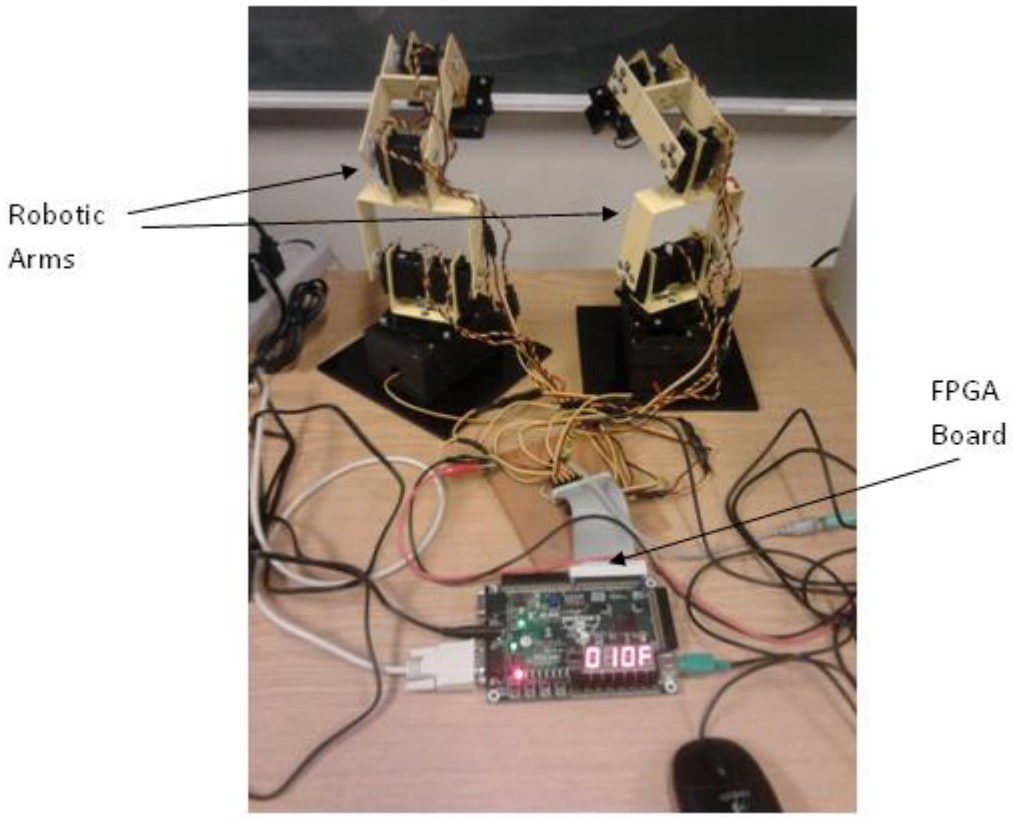


Figure 2.9: Robot assembly

a neutral midpoint position (0°). A 2-ms pulse rotates the shaft to the extreme CW position (+45°). The pulse width is forwarded to the servomotor roughly 50 times per second (50Hz). In our case, a high torque servo (HS-645MG, Figure 2.11) is used for the robotic arms.

The HS-625MG is a new batch of Hitec servos. Utilizing the unique MP and Alumite gear train technology the HS-625 has the infamous guaranteed to be un-breakable gear train.

The powerful HS-625MG is complete for the applications with high demand requiring a standard sized servo. Some specifications are given as [38]:

Control System: +Pulse Width Control.

Operating Speed: 0.20 sec.

Output Torque: $\sim$ 0.8-1 N.m.

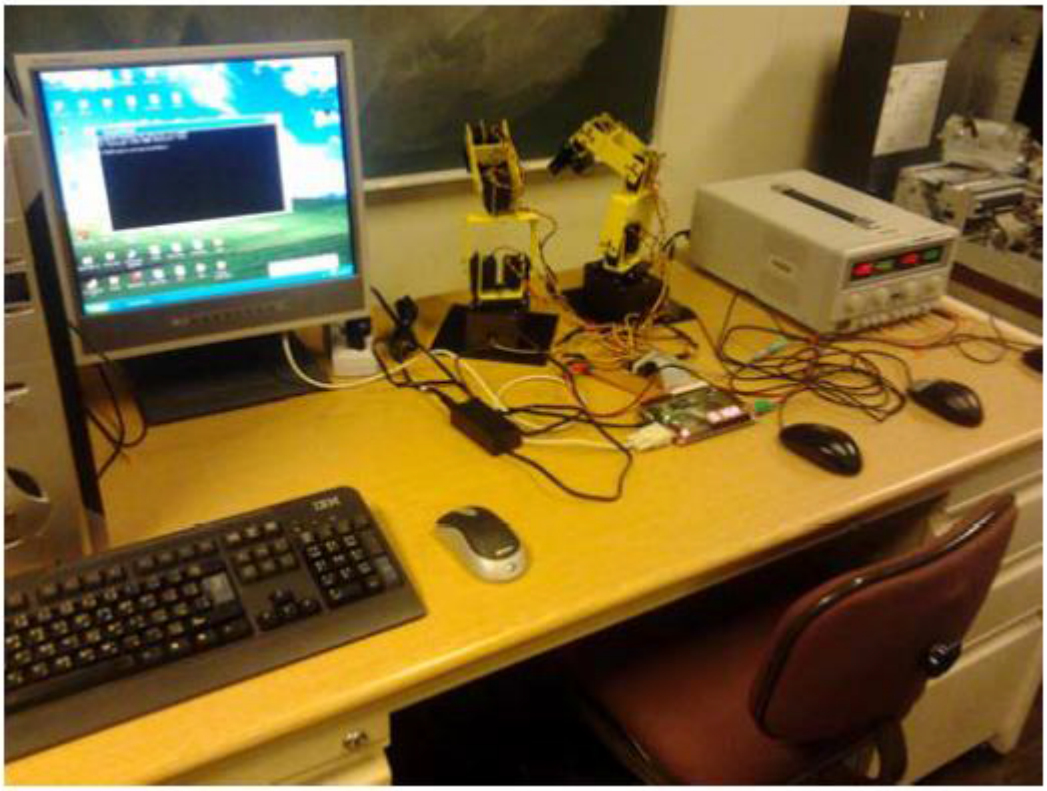


Figure 2.10: Dual robotic arm controller system

Figure 2.11: HS-645MG ultra torque servo

Size: 41x20x38 mm.

Weight: $\sim$ 55 gr.

Required Pulse: 3-5 Volt Peak to Peak Square Wave.

Operating Voltage: 4.8-6.0 Volts. The robotic arms can be programmed through the FPGA board (as shown if figure 2.9) to grab an object from a defined position. The complete set-up of the assembly as shown before (figure 2.9) includes 2 identical robotic arms and an FPGA board. Some limitations for the robotic arms are given in table (2.2).

Table 2.2: Limitations for the robotic arms

Base rotation	$\sim$ 160 degrees
Shoulder moving range	$\sim$ 160 degrees
Elbow moving range	$\sim$ 160 degrees
Wrist moving range	$\sim$ 160 degrees
Gripper open and close	$\sim$ 3.2 cm
Height	$\sim$ 40 cm
Maximum Horizontal Reach	$\sim$ 30 cm

CHAPTER 3

ROBOT DYNAMICS

In this chapter we derive the dynamic model of the robot based on the Euler-Lagrange formulation. To simplify the analysis, dynamic models of two- and three-link planar robots are first derived, followed by the dynamic model development of the actual robotic arm with 4 DOF.

3.1 Introduction to Robot Dynamics

In this section, dynamic models of two- and three-link planar manipulators are derived. The purpose of these developments is to pave the way for the dynamic modeling of the actual 4-DOF robotic manipulator.

3.1.1 Two Link Robot Dynamics

A two-link planar robot having two links of masses m_1 and m_2 and two servos with masses M_1 and M_2 is shown in figure (3.1). The dynamic model of this manipulator can be obtained using the Lagrange equation [39],as follows:

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{q}_i} - \frac{\partial K}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \qquad i = 1, 2, ...n \tag{3.1}$$

where Q_i is the generalized force corresponding to the generalized coordinate q_i. K and V are the total kinetic and potential energies of the system, respectively and n is

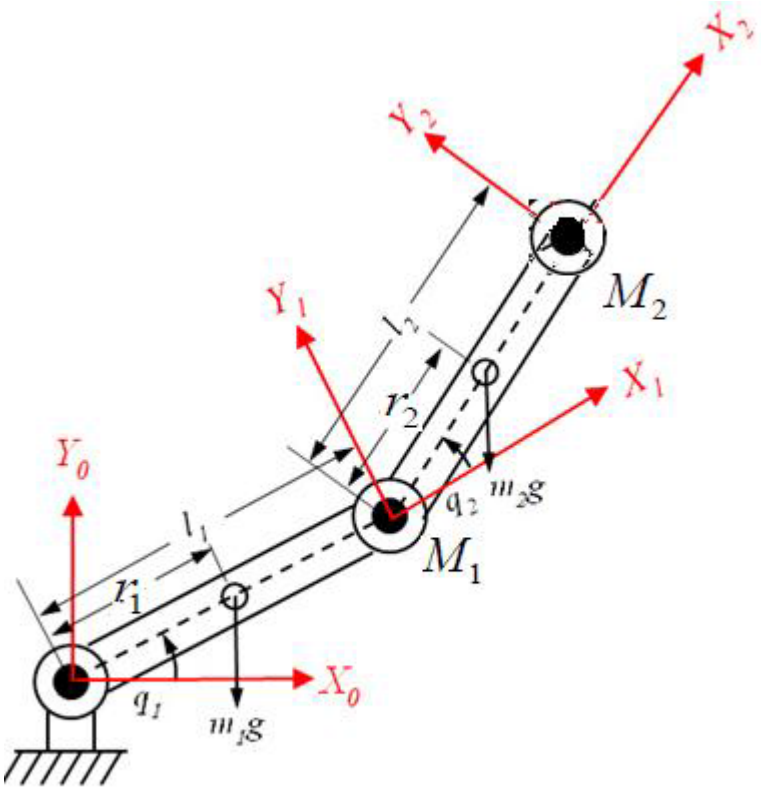


Figure 3.1: Two Links Robotic arm

the number of the generalized coordinates. Some parameters of this manipulator are explained in table (3.1).

Table 3.1: Manipulator Parameters

Parameter	Description
ω_i	Angular velocity of link (i)
v_i	Translational velocity at the end of link (i)
v_{Gi}	Translation velocity at link (i) center of gravity
x_{Gi},y_{Gi}	coordinates of the centre of gravity of link i
x_i,y_i	Global corrdinates at the end of link (i)
I_i	Link (i) moment of inertia

$$x_1 = l_1 \cos\theta_1 \qquad y_1 = l_1 \sin\theta_1 \qquad x_{G1} = r_1 \cos\theta_1 \qquad y_{G1} = r_1 \sin\theta_1$$

$$x_{G2} = l_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2) \qquad y_{G2} = l_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2)$$

$$x_2 = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \qquad y_2 = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

Thus the linear velocities are given as:

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = l_1^2\dot{\theta}_1^2 \qquad v_{G1}^2 = \dot{x}_{G1}^2 + \dot{y}_{G1}^2 = r_1^2\dot{\theta}_1^2 \tag{3.2}$$

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = l_1^2\dot{\theta}_1^2 + 2l_1l_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 + l_2^2(\dot{\theta}_1 + \dot{\theta}_2) \tag{3.3}$$

$$v_{G2}^2 = \dot{x}_{G2}^2 + \dot{y}_{G2}^2 = l_1^2\dot{\theta}_1^2 + 2l_1r_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 + r_2^2(\dot{\theta}_1 + \dot{\theta}_2) \tag{3.4}$$

Having the above parameters, the kinetic energy could be obtained as follows:

$$K = \sum \frac{1}{2} m_i v_{Gi}^2 + \frac{1}{2} M_i v_i^2 + \frac{1}{2} I_i \omega_i^2 \tag{3.5}$$

and the potential energy is given by:

$$\begin{aligned} V =& m_1 g r_1 \sin\theta_1 + m_2 g(l_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2)) + M_1 g l_1 \sin\theta_1 \\ &+ M_2 g(l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)) \end{aligned} \tag{3.6}$$

where in this case the servos are treated as point masses at the end of each link and thus contributes only to the translation part of the kinetic energy (by the means of parallel axes theorem), and have no effect on the rotational one. Substituting (3.5 and 3.6) by taking the appropriate derivatives in (3.1) we get:

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = Q \tag{3.7}$$

The matrix D is the inertia matrix , C is the Coriolis matrix and g is vector of gravitational forces ; they the form:

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}, \quad and \quad g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

Where

$$D_{11} = m_1 r_1^2 + I_1 + m_2(l_1^2 + r_2^2 + 2l_1 r_2 \cos\theta_2) + M_1 l_1^2 + I_2 + M_2(l_1^2 + l_2^2 + 2l_1 l_2 \cos\theta_2)$$

$$D_{12} = m_2(l_1^2 + r_2^2 + 2l_1 r_2 \cos\theta_2) + I_2 + M_2(l_1^2 + l_2^2 + 2l_1 l_2 \cos\theta_2)$$

$$D_{21} = D_{12} \qquad D_{22} = m_2 r_2^2 I_2 + M_2 l_2^2$$

$$C_{11} = -2(m_2 l_1 r_2 \sin\theta_2 + M_2 l_1 l_2 \cos\theta_2)\dot{\theta}_2 \qquad C_{12} = \frac{1}{2} C_{11}$$

$$C_{12} = (m_2 l_1 r_2 \sin\theta_2 + M_2 l_1 l_2 \cos\theta_2)\dot{\theta}_1$$

$$g_1 = m_1 g r_1 \cos\theta_1 + m_2 g(l_1 \cos\theta_1 + r_2 \cos(\theta_1 + \theta_2)) + M_1 g l_1 \cos\theta_1$$
$$+ M_2 g(l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2))$$

$$g_2 = m_2 g r_2 \cos(\theta_1 + \theta_2) + + M_2 g r_2 \cos(\theta_1 + \theta_2)$$

3.1.2 Three Link Robot Dynamics

A three-link planar robot, having links of masses m_1 , m_2 and m_3 and three-servos having masses M_1 , M_2 and M_3 located at each joint is shown in figure 3.2 . Following

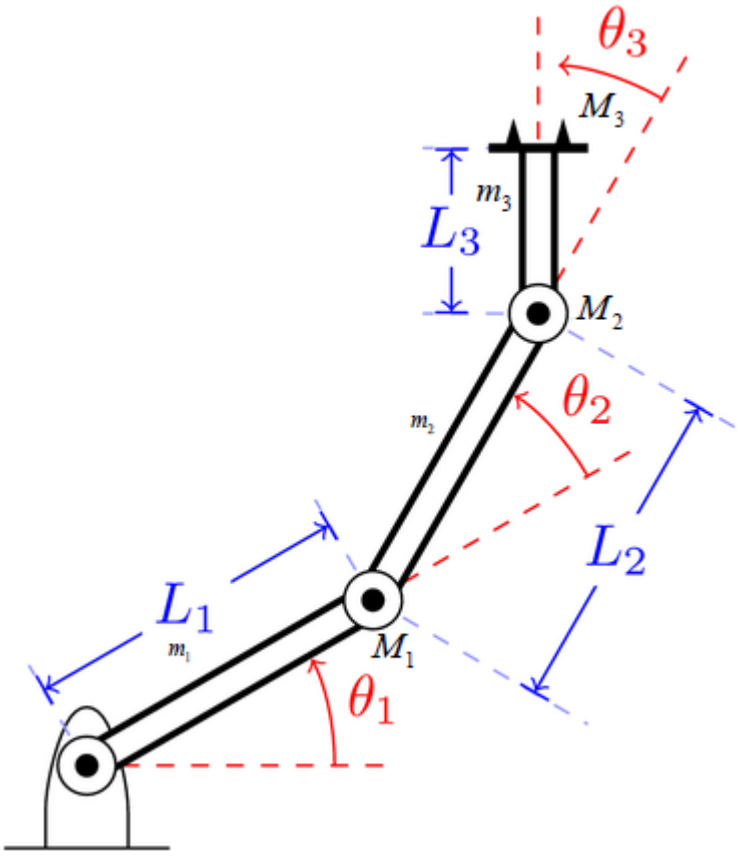


Figure 3.2: Three Links Robotic arm

the same approach as for the case of the two-link robot arm, we get:

the linear velocities v_1, v_2 , v_{G1} and v_{G2} similar to those in (3.2 and 3.3), however, as for v_3 and v_{G3} they take the form:

$$v_3^2 = (l_3 c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) + l_1 \dot{\theta}_1 \cos\theta_1)^2 + (l_3 s_{123}(\dot{\theta}_1 + \dot{\theta}_2) + l_1 \dot{\theta}_1 s_1)^2$$

$$v_{G3}^2 = (r_3 c_{123}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) + l_1 \dot{\theta}_1 \cos\theta_1)^2 + (r_3 s_{123}(\dot{\theta}_1 + \dot{\theta}_2) + l_1 \dot{\theta}_1 s_1)^2$$

The potential energy is given by:

$$\begin{aligned}V =& gm_2(l_1 \sin\theta_1 + r_2 \sin(\theta_1 + \theta_2)) + gm_3(l_1 \sin\theta_1 + l_2 sin(\theta_1 + \theta_2) + r_3 \sin(\theta_1 \\ &+ \theta_2 + \theta_3)) + M_2 g(l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)) + M_3 g(l_1 \sin\theta_1 \\ &+ l_3 \sin(\theta_1 + \theta_2 + \theta_3) + l_2 \sin(\theta_1 + \theta_2)) + l_1 M_1 g \sin\theta_1 + g m_1 r_1 \sin\theta_1\end{aligned}$$

To streamline notations, let:

$$s_i = \sin\theta_i \qquad c_{ijk} = \cos(\theta_i + \theta_j + \theta_k), \quad and \quad s_{iij} = \sin(2\theta_i + \theta + _j) \tag{3.8}$$

The kinetic energy is defined as before (3.5). Substituting these in the Lagrange equation (3.1), we get an equation similar to that (3.7), with matrices defined by:

$$\begin{aligned}D_{11} =& I_1 + I_2 + I_3 + l_1^2 M_1 + l_1^2 M_2 + l_1^2 M_3 + l_2^2 M_2 + l_2^2 M_3 + l_3^2 M_3 + l_1^2 m_2 + l_1^2 m_3 \\ &+ l_2^2 m_3 + m_1 r_1^2 + m_2 r_2^2 + m_3 r_3^2 + 2 l_1 l_2 M_2 \cos\theta_2 + 2 l_1 l_2 M_3 \cos\theta_2 \\ &+ 2 l_2 l_3 M_3 \cos\theta_3 + 2 l_1 m_3 r_3 \cos(\theta_2 + \theta_3) + 2 l_1 l_2 m_3 \cos\theta_2 + 2 l_1 m_2 r_2 \cos\theta_2 \\ &+ 2 l_2 m_3 r_3 \cos\theta_3 + 2 l_1 l_3 M_3 \cos(\theta_2 + \theta_3)\end{aligned}$$

$$\begin{aligned}D_{12} =& I_2 + I_3 + l_2^2 M_2 + L_2^2 M_3 + L_3^2 M_3 + L_2^2 m_3 + m_2 r_2^2 + m_3 r_3^2 + l_1 l_2 M_2 \cos\theta_2 \\ &+ l_1 l_2 M_3 \cos\theta_2 + 2 l_2 l_3 M_3 \cos\theta_3 + l_1 m_3 r_3 \cos(\theta_2 + \theta_3) + l_1 l_2 m_3 \cos\theta_2 \\ &+ l_1 m_2 r_2 \cos\theta_2 + 2 l_2 m_3 r_3 \cos\theta_3 + l_1 l_3 M_3 \cos(\theta_2 + \theta_3)\end{aligned}$$

$$\begin{aligned}D_{13} =& I_3 + l_3^2 M_3 + m_3 r_3^2 + l_2 l_3 M_3 \cos\theta_3 + l_1 m_3 r_3 \cos(\theta_2 + \theta_3) + l_2 m_3 r_3 \cos\theta_3 \\ &+ l_1 l_3 M_3 \cos(\theta_2 + \theta_3)\end{aligned}$$

$$
\begin{aligned}
D_{21} =& I_2 + I_3 + l_2^2 M_2 + L_2^2 M_3 + L_3^2 M_3 + L_2^2 m_3 + m_2 r_2^2 + m_3 r_3^2 + l_1 l_2 M_2 \cos\theta_2 \\
&+ l_1 l_2 M_3 \cos\theta_2 + 2 l_2 l_3 M_3 \cos\theta_3 + l_1 m_3 r_3 \cos(\theta_2 + \theta_3) + l_1 l_2 m_3 \cos\theta_2 \\
&+ l_1 m_2 r_2 \cos\theta_2 + 2 l_2 m_3 r_3 \cos\theta_3 + l_1 l_3 M_3 \cos(\theta_2 + \theta_3)
\end{aligned}
$$

$$
\begin{aligned}
D_{22} =& I_2 + I_3 + l_2^2 M_2 + l_2^2 M_3 + l_3^2 M_3 + l_2^2 m_3 + m_2 r_2^2 + m_3 r_3^2 + 2 l_2 l_3 M_3 \cos\theta_3 \\
&+ 2 l_2 m_3 r_3 \cos\theta_3
\end{aligned}
$$

$$
D_{23} = M_3 l_3^2 + l_2 M_3 \cos\theta_3 l3 + m_3 r_3^2 + l_2 m_3 \cos\theta_3 r_3 + I_3
$$

$$
\begin{aligned}
D_{31} =& I_3 + l_3^2 M_3 + m_3 r_3^2 + l_2 l_3 M_3 \cos\theta_3 + l_1 m_3 r_3 \cos(\theta_2 + \theta_3) + l_2 m_3 r_3 \cos\theta_3 \\
&+ l_1 l_3 M_3 \cos(\theta_2 + \theta_3)
\end{aligned}
$$

$$
D_{32} = M_3 l_3^2 + l_2 M_3 \cos\theta_3 l3 + m_3 r_3^2 + l_2 m_3 \cos\theta_3 r_3 + I_3
$$

$$
D_{33} = M_3 l_3^2 + m_3 r_3^2 + I_3
$$

The Coriolis matrix components are given as:

$$
\begin{aligned}
C_{11} =& -2 l_1 l_3 M_3 \dot{\theta}_2 s_{23} - 2 l_1 l_3 M_3 \dot{\theta}_3 s_{23} - 2 l_1 l_2 M_2 \dot{\theta}_2 s_2 - 2 l_1 l_2 M_3 \dot{\theta}_2 s_2 - 2 l_2 l_3 M_3 \dot{\theta}_3 s_3 \\
&- 2 l_1 \dot{\theta}_2 m_3 r_3 s_{23} - 2 l_1 \dot{\theta}_3 m_3 r_3 s_{23} - 2 l_1 l_2 \dot{\theta}_2 m_3 s_2 - 2 l_1 \dot{theta}_2 m_2 r_2 s_2 \\
&- 2 l_2 \dot{\theta}_3 m_3 r_3 s_3
\end{aligned}
$$

$$C_{12} = -2l_1l_3M_3\dot{\theta}_1s_{23} - l_1l_3M_3\dot{\theta}_2s23 - 2l_1l_3M_3\dot{theta}_3s_{23} - 2l_1l_2M_2\dot{\theta}_1s_2 - l_1l_2M_2\dot{\theta}_2s_2$$
$$- 2l_1l_2M_3\dot{\theta}_1s_2 - l_1l_2M_3\dot{\theta}_2s_2 - 2l_2l_3M_3\dot{\theta}_3s_3 - 2l_1\dot{\theta}_1m_3r_3s_{23} - l_1\dot{\theta}_2m_3r_3s_{23}$$
$$- 2l_1\dot{\theta}_3m_3r_3s_{23} - 2l_1l_2\dot{\theta}_1m_3s_2 - l_1l_2\dot{\theta}_2m_3s_2 - 2l_1\dot{theta}_1m_2r_2s_2 - l_1\dot{\theta}_2m_2r_2s_2$$
$$- 2l_2\dot{\theta}_3m_3r_3s_3$$

$$C_{13} = -(Ll_1\sin(\theta_2+\theta_3) + l_2\sin(\theta_3))(m_3r_3 + l_3M_3)(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)$$

$$C_{21} = l_1l_3M_3\dot{\theta}_1s_{23} + l_1l_2M_2\dot{\theta}_1s_2 + l_1l_2M_3\dot{\theta}_1s_2 - 2l_2l_3M_3\dot{\theta}_3s_3 + l_1\dot{theta}_1m_3r_3s_{23}$$
$$+ l_1l_2\dot{\theta}_1m_3s_2 + l_1\dot{\theta}_1m_2r_2s_2 - 2l_2\dot{\theta}_3m_3r_3s_3$$

$$C_{22} = -2l_2\dot{\theta}_3\sin\theta_3(m_3r_3 + l_3M_3)$$

$$C_{23} = -l_2\sin\theta_3(m_3r_3 + l_3M_3)(2\dot{\theta} + 2\dot{\theta}_2 + \dot{\theta}_3)$$

$$C_{31} = (m_3r_3 + l_3M_3)(l_1\dot{\theta}_1\sin(\theta_2+\theta_3) + l_2\dot{\theta}_1\sin\theta_3 + 2l_2\dot{\theta}_2\sin\theta_3)$$

$$C_{32} = l_2\sin\theta_3(2\dot{\theta}_1 + \dot{\theta}_2)(m_3r_3 + l_3M_3)$$

$$C_{33} = 0$$

Finally, the vector of gravity forces is given by:

$$g_1 = gm_3(l_2\cos(\theta_1+\theta_2) + r_3\cos(\theta_1+\theta_2+\theta_3) + l_1\cos\theta_1 + M_2g(l_2\cos(\theta_1+\theta_2)$$
$$+ l_1\cos\theta_1) + M_3g(l_3\cos(\theta_1+\theta_2+\theta_3) + l_2\cos(\theta_1+\theta_2) + l_1\cos\theta_1)$$
$$+ gm_2(r_2\cos(\theta_1+\theta_2) + l_1\cos\theta_1) + l_1M_1g\cos\theta_1 + gm1r1\cos\theta_1$$

$$g_2 = gm_3(l_2\cos(\theta_1+\theta_2)+r_3\cos(\theta_1+\theta_2+\theta_3))+M_3g(l_3\cos(\theta_1+\theta_2+\theta_3)$$
$$+l_2\cos(\theta_1+\theta_2))+gm_2r_2cos(\theta_1+\theta_2)+l_2M_2g\cos(\theta_1+\theta_2)$$

$$g_3 = gm_3r_3cos(\theta_1+\theta_2+\theta_3)+l_3M_3gcos(\theta_1+\theta_2+\theta_3)$$

To this end the dynamics of the three-link planar robot is derived.

In (3.3) a four-link robotic manipulator, having links of masses m_1, m_2, m_3 and m_4 and four servos at the appropriate locations to manipulate the joints, is depicted.

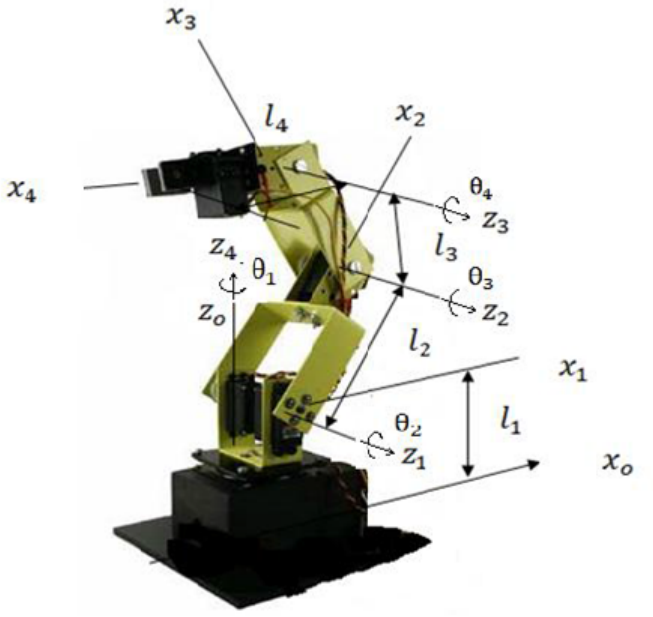


Figure 3.3: Four links robotic arms

3.2 Dynamic Model of the 4 DOF manipulator

In this section we derive the Lagrange equations for the robotic manipulator depicted in figure (3.3). After finding the potential and kinetic energies, the Lagrange equation

(3.9) [40] , can be used to determine the dynamics model of the robotic manipulator. There are four links in this case, and hence four generalized coordinates and consequently four equations.

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{q}} - \frac{\partial K}{\partial q} + \frac{\partial V}{\partial q} = Q \tag{3.9}$$

where K is the kinetic energy, V is the potential energy, q is the generalized coordinate, and Q is the corresponding generalized force.

Introducing the *Lagrangian* L as

$$L(q, \dot{q}) = K(q, \dot{q}) - V(q)$$

one could write the Lagrange equation (3.9), in terms of *Lagrangian* as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q \tag{3.10}$$

3.2.1 Manipulator Lagrangian

To determine the kinetic energy of a robot manipulator (treated as an open-chain mechanism) having n joints, we add the kinetic energy of individual links. We introduce a coordinate system L_i attached at the mass center of the ith link, for this purpose, Let

$$g_{sl_i}(\theta) = e^{\hat{\zeta}\theta_1} e^{\hat{\zeta}\theta_2} ... e^{\hat{\zeta}\theta_n} g_{sl_i}(0) \tag{3.11}$$

represent the configuration of the $L_i$frame with respect to the robot base frame S. The body velocity for the mass center of the link (i) is obtained from

$$V^b_{sl_i} = J^b_{sl_i}(\theta)\dot{\theta}$$

where $J^b_{sl_i}$ is the body Jacobian corresponding to g_{sl_i} and is given by

$$J^b_{sl_i}(\theta) = [\zeta_1^\dagger \quad ...\zeta_i^\dagger \quad 0...0] \tag{3.12}$$

where $\zeta_j^\dagger$is the jth instantaneous joint twist with respect to the frame of the ith link given by

$$\zeta_j^\dagger = Ad^{-1}(e^{\hat{\zeta}_j\theta_j}...e^{\hat{\zeta}\theta_i}...e^{\hat{\zeta}\theta_n}g_{sl_i}(0))\zeta_j \qquad j \leq i \tag{3.13}$$

where the adjoint map gives [41]

$$Ad^{-1}g = \begin{bmatrix} R^T & -R^T\hat{p} \\ 0 & R^T \end{bmatrix} \tag{3.14}$$

in which R is the rotational part of the matrix g and $\hat{p}$ is the skew-symmetric matrix generated from the position vector p of the rotation matrix g. To simplify the analysis, we write the Jacobian $J^b_{sl_i}$ as J_i. The ith link kinetic energy is

$$K(\theta,\dot{\theta}) = \frac{1}{2}(V^b_{sl_i})^T\mathcal{M}_iV^b_{sl_i} = \frac{1}{2}\dot{\theta}^TJ_i^T\mathcal{M}_iJ_i\dot{\theta} \tag{3.15}$$

where $\mathcal{M}$ is the generalized matrix of inertia of the link (i). Thus, the overall expression for the total kinetic energy becomes

$$K(\theta,\dot{\theta}) = \sum_{i=1}^{n} K_i(\theta,\dot{\theta}) = \frac{1}{2}\dot{\theta}^TD(\theta)\dot{\theta} \tag{3.16}$$

where the matrix$D(\theta) \in R^{n\times n}$ is the matrix of inertia of the manipulator. In terms of the link Jacobian, J_i , the inertia matrix of the robot manipulator is given by

$$D(\theta,\dot{\theta}) = \sum_{i=1}^{n} J_i^T\mathcal{M}_iJ_i \tag{3.17}$$

Next, we need to know the potential energy of the robot, for which we introduce h_i as the height for the mass center of the ith link. Then the potential energy is given by

$$V_i(\theta) = m_i g h_i(\theta)$$

Thus the Lagrangian yields

$$L(\theta, \dot{\theta}) = \sum_{i=1}^{n} (K(\theta, \dot{\theta}) - V_i(\theta)) = \frac{1}{2}\dot{\theta}^T D(\theta)\dot{\theta} - V(\theta) \tag{3.18}$$

3.2.2 General expression for the equations of motion

The Lagrangian

$$L(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T D(\theta)\dot{\theta} - V(\theta)$$

where D is the matrix of inertia of the manipulator and V is the potential energy because of gravitational forces. It would be worthy to write the kinetic energy in summation form, that is

$$L(\theta, \dot{\theta}) = \frac{1}{2}\sum_{i=1}^{n} D_{ij}(\theta)\dot{\theta}_i\dot{\theta}_j - V(\theta)$$

Substituting in the equation of motion

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q$$

where Q_i represents, in addition to the torque of the actuator, the nonconservative, generalized forces acting on the ith link. One gets

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = \frac{d}{dt}(D_{ij}(\theta)\dot{\theta}_j) = \sum_{i=1}^{n}(D_{ij}(\theta)\ddot{\theta}_j + \dot{D}_{ij}(\theta)\dot{\theta}_j)$$

Using the chain rule of differentiation, the term $\dot{D}_{ij}$, could be factored to obtain, for the Lagrange equation [30]

$$\sum_{j=1}^{n} D_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k=1}^{n} (\frac{\partial D_{kj}}{\partial \theta_k}\dot{\theta}_j\dot{\theta}_k - \frac{1}{2}\frac{\partial D_{kj}}{\partial \theta_i}\dot{\theta}_j\dot{\theta}_j) + \frac{\partial V}{\partial \dot{\theta}_i}(\theta) = Q_i \qquad i = 1, ..., n \quad (3.19)$$

Rewriting the terms, we get

$$\sum_{j=1}^{n} D_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k=1}^{n} \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k + \frac{\partial V}{\partial \dot{\theta}_i}(\theta) = Q_i \qquad i = 1, ..., n \quad (3.20)$$

where the functions Γ_{ijk} denote the Christoffel symbols related to the Inertia matrix $D(\theta)$ given by:

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial D_{ij}}{\partial \theta_k} + \frac{\partial D_{ik}}{\partial \theta_j} - \frac{\partial D_{kj}}{\partial \theta_i}\right) \quad (3.21)$$

Equation (3.21) is a differential equation of the second order in terms of the joint angles of manipulator . It is made up of four parts: inertial forces, that rely on the acceleration of the robot joints (second derivative of the generalized coordinates), Coriolis and centrifugal forces quadratic in the joint speeds, gravitational (potential) forces, of the form $\frac{\partial V}{\partial \theta}$, and exterior forces Q_i . In the classical literature of mechanics , the terms of the form $\dot{\theta}_i\dot{\theta}_j$ are identified as Coriolis forces and those of the form $\dot{\theta}_k^2$ as centrifugal forces.

The exterior forces can be partitioned in two parts. Let τ represent the joint applied torque and introduce $-N(\theta, \dot{\theta})$ to include any extra forces working on the ith generalized coordinate, such as conservative forces emerging from a frictional and potential forces. For example, if the robot has joints with viscous friction, then N_i would be defined as [30]:

$$-N(\theta, \dot{\theta}) = -\frac{\partial V}{\partial \theta_i} - \beta(\dot{\theta}_i)$$

where β represents the damping coefficient.

3.2.3 Model in Compact Form

For control requirements, it is common and more feasible to rephrase the Lagrangian dynamic model of the manipulator in the compact form (in a matrix form) as below

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}) = \tau \tag{3.22}$$

Hereafter, assuming that the external forces are restricted to the applied joint torques in addition to the conservative forces arising from the gravity (potential energy), the term $N(\theta,\dot{\theta})$ is replaced by $g(\theta)$ and Eq (3.22) is written as:

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + g(\theta) = \tau \tag{3.23}$$

The matrix $C(\theta,\dot{\theta}) \in R^{n\times n}$ is called the Coriolis for the robot, the vector $C(\theta,\dot{\theta})\dot{\theta}$ represents the terms of centrifugal and Coriolis forces in the equations of motion. The matrix C is given by

$$C(\theta,\dot{\theta}) = \Gamma_{ijk}\dot{\theta}_k = \frac{1}{2}\left(\frac{\partial D_{ij}}{\partial \theta_k} + \frac{\partial D_{ik}}{\partial \theta_j} - \frac{\partial D_{kj}}{\partial \theta_i}\right)\dot{\theta}_k \tag{3.24}$$

where τ is the vector of actuator torques, and $g(\theta)$ is the vector of gravitational forces. Equation (3.23) represents a second-order differential equation for the robot's motion as a function of the joint applied torques in a vector form.

Referring to the matrix form of equation (3.22) in the case of a four degrees of freedom, the inertia matrix $D(\theta)$ is 4×4 and the other matrices are in the form of 4×1 vectors.

$$D(\theta) = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix}, C(\theta,\dot{\theta})\dot{\theta} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix}, g(\theta) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}, \text{and} \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}$$

To find the inertia matrix, using the equation (3.17), we need to know the body Jacobian for every link. By attaching a coordinate frame at the mass center of each link, the generalized inertia matrix simplifies to the diagonal matrix as below

$$\mathcal{M} = \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xi} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yi} & 0 \\ & 0 & 0 & 0 & 0 & I_{zi} \end{bmatrix} \tag{3.25}$$

Next we need to find the body Jacobian. Using equations (3.12) and (3.13), we get

$$J_1 = J^b_{sl_1(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \qquad J_2 = J^b_{sl_2(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ r_2 \cos\theta_2 & 0 & 0 & 0 \\ 0 & r_2 & 0 & 0 \\ \sin\theta_2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ \cos\theta_2 & 0 & 0 & 0 \end{bmatrix}$$

$$J_3 = J^b_{sl_3(0)} = \begin{bmatrix} 0 & l_2 s_3 & 0 & 0 \\ r_3 c_{23} + l_2 c_2 & 0 & 0 & 0 \\ 0 & r_3 + l_2 c_3 & r_3 & 0 \\ s_{23} & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ c_{23} & 0 & 0 & 0 \end{bmatrix}$$

$$J_4 = J^b_{sl_4(0)} = \begin{bmatrix} 0 & l_2 s_{34} + l_3 s_4 & l_3 s_4 & 0 \\ l_3 c_{23} + l_2 c_2 + r_4 c_{234} & 0 & 0 & 0 \\ 0 & r_4 + l_2 c_{34} + l_3 c_4 & r_4 + l_3 c_4 & r_4 \\ s_{234} & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ c_{234} & 0 & 0 & 0 \end{bmatrix}$$

The inertia matrix is given by

$$D(\theta) = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} = J_1^T \mathcal{M}_1 J_1 + J_2^T \mathcal{M}_2 J_2 + J_3^T \mathcal{M}_3 J_3 + J_4^T \mathcal{M}_4 J_4 \quad (3.26)$$

The elements of D are calculated as below

$$\begin{aligned} D_{11} =& I_{z1} + I_{z4} c_{234}^2 + I_{x4} s_{234}^2 + m_4 (L_3 c_{23} + L_2 c_2 + r_4 c_{234})^2 + m_3 (r_3 c_{23} + L_2 c_2)^2 \\ &+ I_{z3} c_{23}^2 + I_{x3} s_{23}^2 + I_{z2} c_2^2 + I_{x2} s_2^2 + m_2 r_2^2 c_2^2 \end{aligned}$$

$$D_{12} = 0, \quad D_{13} = 0, \quad and \quad D_{14} = 0 \qquad D_{21} = 0 \quad (3.27)$$

$$\begin{aligned} D_{22} =& I_{y2} + I_{y3} + I_{y4} + m_4 (L_2 s_{23} + L_3 s_4)^2 + m_3 (r_3 + L_2 c_3)^2 + m_2 r_2^2 \\ &+ m_4 (r_4 + L_2 c_{34} + L_3 c_4)^2 + L_2^2 m_3 s_3^2 \end{aligned}$$

$$\begin{aligned} D_{23} =& I_{y3} + I_{y4} + m_3 r_3 (r_3 + L_2 c_3) + m_4 (r_4 + L_3 c_4)(r_4 + L_2 c_{34} + L_3 c_4) \\ &+ L_3 m_4 s_4 (L_2 s_{23} + L_3 s_4) \end{aligned}$$

$$D_{24} = I_{y4} + m_4 r_4 (r_4 + L_2 c_{34} + L_3 c_4)$$

$$D_{33} = I_{y3} + I_{y4} + m_4 (r_4 + L_3 c_4)^2 + m_3 r_3^2 + L_3^2 m_4 s_4^2$$

$$D_{34} = I_{y4} + m_4 r_4 (r_4 + L_3 c_4) \qquad D_{31} = 0 \quad D_{32} = D_{23}$$

$$D_{41} = 0, \quad D_{42} = D_{24}, \quad D_{43} = D_{34}, \quad and \quad D_{44} = m_4 r_4^2 + I_{y4}$$

Having the inertia matrix in hand, the centrifugal forces and Coriolis are directly determined from equation (3.24). It could be shown that the nonzero terms of the Christoffel symbols Γ_{ijk} equation (3.21) are given as follows

$$\begin{aligned}\Gamma_{112} =& (I_{x3} - I_{z3} - m_3 r_3^2 + m_4 l_3^2)(s_{23} c_{23}) + (I_{x2} - I_{z2})(s_2 c_2) + (I_{x4} - I_{z4})(s_{234} c_{234}) - \\ & (m_2 r_2^2 + m_3 l_2^2 + m_4 l_2^2)(s_2 c_2) - m_4 r_4 (\frac{1}{2} r_4 + l_2 + l_3) \sin(2\theta_2 + \theta_3 + \theta_4) \\ & - (m_3 l_2 r_3 + m_4 l_2 l_3) \sin(2\theta_2 + \theta_3)\end{aligned}$$

$$\begin{aligned}\Gamma_{113} =& (I_{x3} - I_{z3})(s_{23} c_{23}) + (I_{x4} - I_{z4})(s_{234} c_{234}) - m_4 (l_3 s_{23} + r_4 s_{234}) \\ & (l_2 c_2 l_3 c_{23} + r_4 c_{234}) - m_3 r_3 s_{23} (l_2 c_2 + r_3 c_{23})\end{aligned}$$

$$\Gamma_{114} = -s_{234}((I_{z4} - I_{x4} + m_4 r_4^2) c_{234} + m_4 l_3 r_4 c_{23 + m_4 l_2 r_4 c_2})$$

$$\Gamma_{121} = \Gamma_{112} \qquad \Gamma_{131} = \Gamma_{113} \qquad \Gamma_{141} = \Gamma_{114}$$

$$\begin{aligned}\Gamma_{211} =&(I_{z3} - I_{x3})s_{23}c_{23} + (I_{z2} - I_{x2})s_2c_2 + (I_{z4} - I_{x4} + m_4r_4^2)s_{234}c_{234} + m_3r_3^2s_{23}c_{23} \\ &+ (m_2r_2^2m_3l_2^2 + m_4l_2^2)l_2^2s_2c_2 + m_4l_3^2s_{23}c_{23}+)m_3l_2r_3 + m_4l_2l_3)s_{223} + m_4r_4l_2s_{2234} \\ &+ m_4r_4l_3s_{22334}\end{aligned}$$

$$\Gamma_{223} = -l_2(m_3r_3s_3 + m_4r_4s_{34} + m_4l_3s_3) \qquad \Gamma_{224} = -m_4r_4(l_2s_{34} + l_3s_4s)$$

$$\Gamma_{232} = -l_2(m_3r_3s_3 + m_4r_4s_{34} + m_4l_3s_3) \qquad \Gamma_{233} = -l_2(m_3r_3s_3 + m_4r_4s_{34} + l_3m_4s_3)$$

$$\Gamma_{242} = -m_4r_4(l_2s_{34} + l_3s_4s) \qquad \Gamma_{243} = -m_4r_4(l_2s_{34} + l_3s_4s)$$

$$\Gamma_{244} = -m_4r_4(l_2s_{34} + l_3s_4s) \qquad \Gamma_{234} = -m_4r_4(l_2s_{34} + l_3s_4)$$

$$\begin{aligned}\Gamma_{311} =&(I_{z3} - I_{x3})s_{23}c_{23}(I_{z4} - I_{x4})s_{234}c_{234} + m_4(l_3s_{23} + r_4s_{234})(l_3c_{23} + l_2c_2 + r_4c_{234})+ \\ &m_3r_3s_{23}(r_3c_{23} + l_2c_2)\end{aligned}$$

$$\Gamma_{322} = l_2m_3r_3s_3 + m_4r_4s_{34} + l_3m_4s_3) \qquad \Gamma_{324} = -l_3m_4r_4s_4 \qquad \Gamma_{334} = -l_3m_4r_4s_4$$

$$\Gamma_{422} = m_4r_4(l_2s_{34} + l_3s_4s) \qquad \Gamma_{344} = -l_3m_4r_4s_4 \qquad \Gamma_{343} = -l_3m_4r_4s_4$$

$$\Gamma_{423} = l_3m_4r_4s_4 \qquad \Gamma_{432} = l_3m_4r_4s_4 \qquad \Gamma433 = l_3m_4r_4s_4$$

$$\Gamma_{411} = s_{234}((I_{z4} - I_{x4} + m_4r_4^2)c_{234} + m_4l_3r_4c_{23+m_4l_2r_4c_2}) \qquad \Gamma342 = -l_3m_4r_4s_4$$

Next, we derive the components of the gravity forces on the robotic arm. These forces can be found from

$$g(\theta) = \frac{\partial V}{\partial \theta_i}$$

Recall that $V : R^n \rightarrow R$ is the robot potential energy and in the case of four-link robotic arm under study, this property is given by

$$V(\theta) = m_1 g h_1(\theta) + m_2 g h_2(\theta) + m_3 g h_3(\theta) + m_4 g h_4(\theta) \tag{3.28}$$

where h_i is the height of the mass center of the link (i), which can be obtained using the direct kinematic formulation presented earlier as

$$g_s l_i(\theta) = e^{\hat{\zeta}\theta_1} e^{\hat{\zeta}\theta_2} ... e^{\hat{\zeta}\theta_n} g_{sl_i}(0) \tag{3.29}$$

Carrying out the required math, we get:

$h_1(\theta) = r_1$

$h_2(\theta) = l_1 + r_2 \sin(\theta_2)$

$h_3 = l_1 + r_3 sin(\theta_2 + \theta_3) + l_2 sin(\theta_2)$

$h_4 = l_1 + l_3 sin(\theta_2 + \theta_3) + l_2 sin(\theta_2) + r_4 sin(\theta_2 + \theta_3 + \theta_4)$

Introducing these expressions into the potential energy and carrying out its derivatives yield:

$$\frac{\partial V}{\partial \theta} = \begin{bmatrix} 0 \\ g(m_3(r_3 c_{23} + l_2 c_2) + m_4(l_3 c_{23} + l_2 c_2 + r_4 c_{234}) + m_2 r_2 c_2) \\ g(m_4(l_3 c_{23} + r_4 c_{234}) + m_3 r_3 c_{23}) \\ g m_4 r_4 c_{234} \end{bmatrix}$$

We highlight that the convenient state variables to describe the manipulator dynamic model are the positions θ_1, θ_2, θ_3, and θ_4, and the velocities $\dot{\theta}_1$ $\dot{\theta}_2$, $\dot{\theta}_3$ and $\dot{\theta}_4$. In

terms of these state variables, the manipulator dynamic model may be written as

$$\frac{d}{dt}\begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \theta_4\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\\ \dot{\theta}_4\\ (M(\theta)^{-1}[\tau(t) - C(\theta,\dot{\theta})\dot{\theta} - g(\theta)]\end{bmatrix}$$

3.3 Properties of the Dynamic model

The dynamic model presented in (3.22) carries some important properties especially in the control of the robot manipulator. The major properties are:

3.3.1 Symmetry in the inertia matrix

The $n\times n$ inertia matrix D is symmetric (and positive definite), i.e. the inertia matrix must satisfy:

$$D = D^T \tag{3.30}$$

The fact that the inertia matrix is symmetric is due to the dynamic interaction between the manipulator links. In other words, there are equal action and reaction forces between the links. Carefully, inspecting the inertia matrix (3.2.3), we can see that as in equation (3.2.3):

$$D_{12} = 0, \quad D_{13} = 0, \quad and \quad D_{14} = 0 \qquad D_{21} = 0 \tag{3.31}$$

There is no dynamic interactions due to acceleration between the first link and the other three links; and this refereed to the physical fact that the first link is rotating in different plane than the other do. However, the interaction between the first link and the other links is only due to the centrifugal and Coriolis terms, which are interpreted

as nonlinear effects due to the configuration-dependent nature of the inertia matrix $D(\theta)$ in Lagrangian formulation.

3.3.2 Skew symmetry property

The matrix E defined by:

$$E(\theta, \dot{\theta}) = \dot{D}(\theta) - 2C(\theta, \dot{\theta}) \tag{3.32}$$

is skew-symmetric matrix, that is, it's elements satisfy $E_{ij} = -E_{ji}$. In other words, the matrix E must satisfy:

$$E + E^T = 0 \tag{3.33}$$

It could be shown that, the off-diagonal elements of the matrix E are given as:

$$\begin{aligned} E_{12} =& \dot{\theta}_1(Iz3s_{2233} - I_{x3}s_{2233} - I_{x2}s_{22} + I_{z2}s_{22} - I_{x4}s_{223344} + I_{z4}s_{223344} \\ &+ m_3r_3^2s_{2233} + l_2^2m_3s_{22} + l_2^2m_4s_{22} + m_2r_2^2s22 + m_4r_4^2s_{223344} + l_3^2m_4s_{2233} + \\ &2l_2m_4r_4s_{2234} + 2l_3m_4r_4s_{22334} + 2l_2l_3m_4s_{223} + 2l_2m_3r_3s_{223}) \end{aligned}$$

$$\begin{aligned} E_{13} =& 2\dot{\theta}_1((Iz3s_{2233})/2 - (Ix3s_{2233})/2 - (Ix4s_{223344})/2 + (I_{z4}s_{223344})/2 \\ &+ m_4(l_3s_{23} + r_4s_{234})(l_3c_{23} + l_2c_2 + r_4c_{234}) + m_3r_3s_{23}(r_3c_{23} + l_2c_2)) \end{aligned}$$

$$\begin{aligned} E_{21} =& -\dot{\theta}_1(Iz3s_{2233} - I_{x3}s_{2233} - I_{x2}s_{22} + I_{z2}s_{22} - I_{x4}s_{223344} + I_{z4}s_{223344} \\ &+ m_3r_3^2s_{2233} + l_2^2m_3s_{22} + l_2^2m_4s_{22} + m_2r_2^2s22 + m_4r_4^2s_{223344} + l_3^2m_4s_{2233} + \\ &2l_2m_4r_4s_{2234} + 2l_3m_4r_4s_{22334} + 2l_2l_3m_4s_{223} + 2l_2m_3r_3s_{223}) \end{aligned}$$

$$
\begin{aligned}
E_{23} =& l_2(2\dot{\theta}_2 m_4 r_4 s_{34} + \dot{\theta}_3 m_4 r_4 s_{34} + \dot{\theta}_4 m_4 r_4 s_{34} \\
&+ 2l_3\dot{\theta}_2 m_4 s_3 + l_3\dot{\theta}_3 m_4 s_3 + 2\dot{\theta}_2 m_3 r_3 s_3 + \dot{\theta}_3 m_3 r_3 s_3)
\end{aligned}
$$

$$
\begin{aligned}
E_{31} =& -2\dot{\theta}_1((I_{z3}s_{2233})/2 - (Ix3s_{2233})/2 - (I_{x4}s_{223344})/2 + (I_{z4}s_{223344})/2 \\
&+ m_4(l_3 s_{23} + r_4 s_{234})(l_3 c_{23} + l_2 c_2 + r_4 c_{234}) + m_3 r_3 s_{23}(r_3 c_{23} + l_2 c_2))
\end{aligned}
$$

$$
\begin{aligned}
E_{32} =& -l_2(2\dot{\theta}_2 m_4 r_4 s_{34} + \dot{\theta}_3 m_4 r_4 s_{34} + \dot{\theta}_4 m_4 r_4 s_{34} \\
&+ 2l_3\dot{\theta}_2 m_4 s_3 + l_3\dot{\theta}_3 m_4 s_3 + 2\dot{\theta}_2 m_3 r_3 s_3 + \dot{\theta}_3 m_3 r_3 s_3)
\end{aligned}
$$

CHAPTER 4

MANIPULATOR CONTROL

Control is the science of desired motion. It relates the dynamics and kinematics of a robot to stipulate motion. It includes optimization issues to determine required input forces and torques so that the system will behave optimally. A classical example is the situation in which the initial and terminal manipulator configurations are given and we need to know the forces acting on the robot to have the motion in minimum time. After an introductory text in control and its importance, this chapter applies two common control approaches to the robotic manipulator under study. The first problem is related to the position control of the manipulator where the initial and the final positions are provided and the second problem is related to the trajectory control, where the whole journey is in concern.

4.1 Introduction to Robot Control

Robotic manipulators belong to a general class of multi-input nonlinear systems, in which feedback control and, in particular, the problems of robustness issues represent important areas of research. In what follows, we review the execution of several control schemes for controlling the robotic arm. In each approach, pros and cons associated are discussed, aiming at demonstrating the physical understanding of the mathematical procedures utilized to develop the controller. Hereby, we introduce first

the fundamental control techniques to establish the basics for physical realization and move on to more complicated approaches.

Modern types of robots are supposed to function rapidly and more precisely. In many situations, to perform a specific task, the gripper of the robot has to track a path accurately. This requires the robust trajectory-tracking for the joint angles. Because the dynamics of the robots is highly nonlinear, it is a non-trivial task to design a controller for them.

4.2 Dynamic Model in Compact Form

As mentioned before, the non linear robot dynamics of a manipulator acts a chief barrier when developing the controller. The robot manipulator dynamic equations define an intricate, nonlinear and multi-variable system. Thus, in this section, the robot control problem is treated in the context of nonlinear and multi-variable control. In what follows, the dynamic model derived earlier is presented in the compact form given by

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \tag{4.1}$$

where $D(\theta)$ is the manipulator inertia matrix, always square, symmetric and positive definite, $C(\theta, \dot{\theta})$ is the Coriolis matrix, $g(\theta)$ is the gravitational force vector and τ is the vector of applied torques.

It is worthy to mention that the matrix D (usually written $D(\theta)$) depends on the joint variable θ. i.e. configuration dependent matrix. Ultimately, a controller needs to be designed to determine the entailed input commands for the actuator such that the joint variables θ and speeds $\dot{\theta}$ track a specified trajectory. The joint variables vector θ and the joint speeds vector $\dot{\theta}$ represent the manipulator's states. Equation (4.1) in standard form, is a second order nonlinear differential equation. Though it is sufficient for the controller design, it is a common practice to write the first order

form by grouping the joint velocities with the joint positions in one vector denoted as the manipulator state vector

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \tag{4.2}$$

"The dynamic state of the manipulator not only depends on the joint positions, as a representative of the geometrical configuration of the manipulator, but also depends on the joint speeds, which reflect the motion of the manipulator" [42].

With this definition, the robot equations of motion can be expressed in a general form of the first-order as [42]

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ M(\theta)^{-1}(\tau - C(\theta, \dot{\theta})\dot{\theta} - g(\theta)) \end{bmatrix} \tag{4.3}$$

4.3 Position Control

In this section the problem of position control is discussed using proportional derivative (PD) control with and without gravity and the proportional derivative integral (PID) control.

4.3.1 PD Control without Gravity Term

A PD-control law can take a vector form

$$u = K_P\tilde{\theta} - K_D\dot{\theta} \tag{4.4}$$

where $\tilde{\theta} = \theta^d - \theta$ is the difference (error) between the required joint position θ^d and the real positions of the joint θ, K_D and K_P are positive diagonal matrices of derivative and proportional gains, respectively. First we show that, in the gravity omission (like a planar robot works in the horizontal plan for example) , that is, if $g(\theta)$ is zero in

equation (4.1), then the PD control law (4.4) attains asymptotic tracking of the joint required displacements. To manifest that the previous control law acquires a steady state error of zero, contemplate the function of Lyapunov in the following form

$$V = \frac{1}{2}\dot{\theta}^T D(\theta)\dot{\theta} + \frac{1}{2}\tilde{\theta}^T K_P \tilde{\theta} \tag{4.5}$$

The first term in (4.5) represents the kinetic energy of the system and the second term acts for the proportional feedback $K_P\tilde{\theta}$. Note that V denotes the total amount of the kinetic energy that would be produced if the actuators of the joints were to be exchanged by springs with stiffnesses denoted by K_P and with positions of equilibrium at θ^d . Thus V is a positive function save at the targeted position i.e. $\theta = \theta^d$, $\dot{\theta} = 0$, where V is zero. Taking the time derivative of V (4.5)

$$\dot{V} = \dot{\theta}^T D(\theta)\ddot{\theta} + \frac{1}{2}\dot{\theta}^T \dot{D}(\theta)\dot{\theta} - \dot{\theta}^T K_P \tilde{\theta} \tag{4.6}$$

By substituting for $D(\theta)\ddot{\theta}$ from (4.3), with $g(\theta) = 0$, it turns out that

$$\begin{aligned} \dot{V} &= \dot{\theta}^T(u - C(\theta,\dot{\theta})) + \frac{1}{2}\dot{\theta}^T \dot{D}(\theta)\dot{\theta} - \dot{\theta}^T K_P \tilde{\theta} \\ &= \dot{\theta}^T(u - K_P\tilde{\theta}) + \frac{1}{2}\dot{\theta}^T(\dot{D}(\theta) - 2C(\theta,\dot{\theta})\dot{\theta} \\ &= \dot{\theta}(u - K_P\tilde{\theta}) \end{aligned} \tag{4.7}$$

where we used the fact that, $\dot{D} - 2C$ is skew symmetric. Finally, using the control law (4.4) in the last equation, we obtain

$$\dot{V} = -\dot{\theta}^T K_D \dot{\theta} \leq 0 \tag{4.8}$$

The previous equation hints that the kinetic energy's rate ($\dot{V}$) of the manipulator

and the virtual spring-dampers, is negative, which indicates that the rate is decreasing with time as long as $\dot{\theta} \neq 0$, until $\dot{\theta}$ reaches zero and the robotic arm terminates at a position of equilibrium, which establishes the system stability. Nevertheless, we have to ensure that the equilibrium final position is the actual required location and the robot does not terminate at a different location, in which V equal 0 while θ does not equal θ^d. This alone is not sufficient to manifest the desired result since it is possible that the robot can reach a location in which $\dot{\theta} = 0$ but $\theta \neq \theta^d$. To prove that this cannot occur we can use LaSalle's Theorem [42]. Assume $\dot{V} = 0$, thus (4.8) yields $\dot{\theta} = 0$ and thus $\ddot{\theta} = 0$. From equation (4.1) and (4.4) with $\tau = u$, without gravity term we get

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = K_P\tilde{\theta} - K_D\dot{\theta} \tag{4.9}$$

Then we must have (the dynamics at the equilibrium):

$$K_P\tilde{\theta} = 0 \tag{4.10}$$

which hints that $\tilde{\theta} = 0$, $\dot{\theta} = 0$. LaSalle's Theorem then entails that the equilibrium is asymptotically stable.

For this type of control we take the robot from an initial position: $\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

To the final desired position: $\theta^d = \begin{bmatrix} \frac{\pi}{3} \\ \frac{\pi}{4} \\ \frac{\pi}{6} \\ \frac{\pi}{12} \end{bmatrix}$ The controller parameters are taken as follows:

$$K_P = \begin{bmatrix} 400 & 0 & 0 & 0 \\ 0 & 600 & 0 & 0 \\ 0 & 0 & 650 & 0 \\ 0 & 0 & 0 & 400 \end{bmatrix}$$

$$K_D = \begin{bmatrix} 400 & 0 & 0 & 0 \\ 0 & 400 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 400 \end{bmatrix}$$

These parameters are chosen after some trials in a way to make the manipulator's joints to reach their final positions in a short period of time, with no oscillation and with reasonable ultimate results. The feedback control used is given as below

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} K_{P1} & 0 & 0 & 0 \\ 0 & K_{P2} & 0 & 0 \\ 0 & 0 & K_{P3} & 0 \\ 0 & 0 & 0 & K_{P4} \end{bmatrix} \begin{bmatrix} \theta_1^d - \theta_1 \\ \theta_2^d - \theta_2 \\ \theta_3^d - \theta_3 \\ \theta_4^d - \theta_4 \end{bmatrix} - \begin{bmatrix} K_{D1} & 0 & 0 & 0 \\ 0 & K_{D2} & 0 & 0 \\ 0 & 0 & K_{D3} & 0 \\ 0 & 0 & 0 & K_{D4} \end{bmatrix} \tag{4.11}$$

By substituting the control law (4.11) in the 1^{st} order differential equation (4.3), integrating the set of differential equations numerically using the Matlab function ODE, we can then simulate the motion of the joints as will be presented shortly. The physical parameters used are listed in up index (5.2)

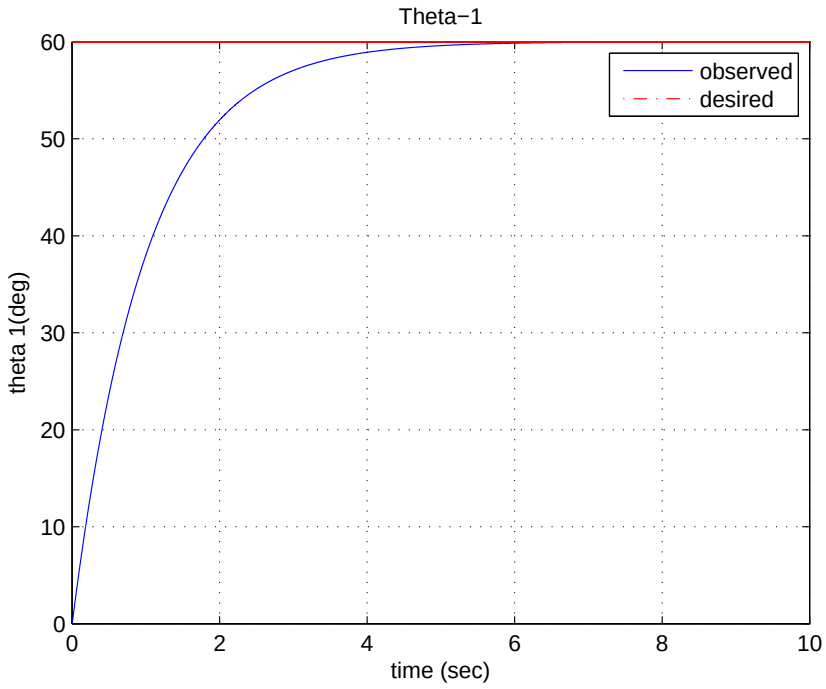


Figure 4.1: Theta 1 Response 'PD control'

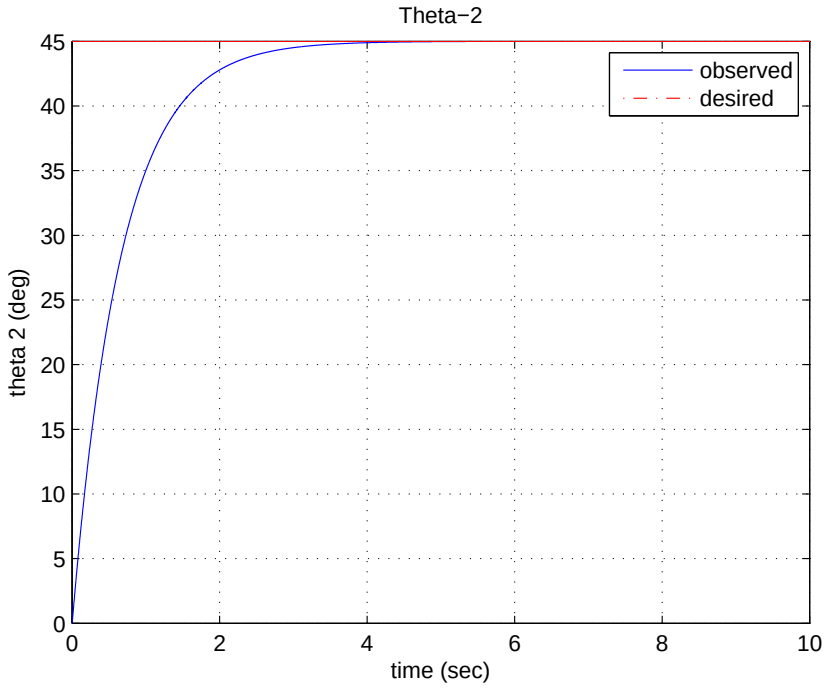


Figure 4.2: Theta 2 Response 'PD control'

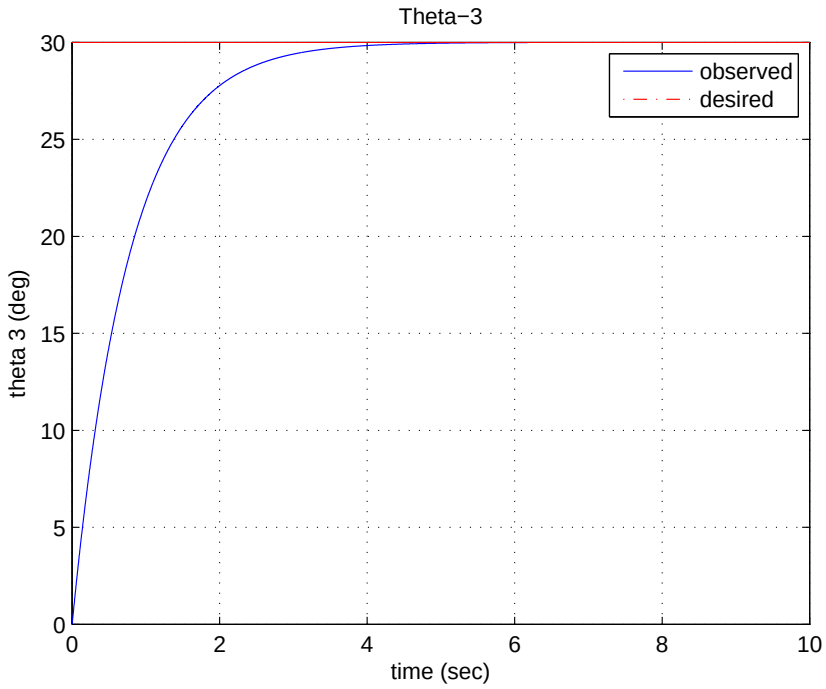


Figure 4.3: Theta 3 Response 'PD control'

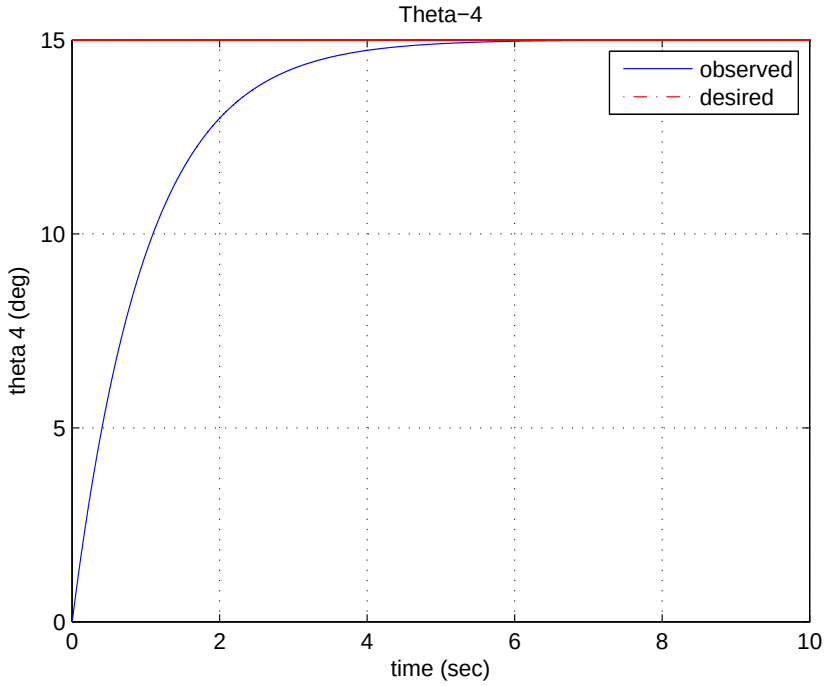


Figure 4.4: Theta 4 Response 'PD control'

For the joint velocities, it is clear from figures 4.5, 4.6, 4.7 and 4.8 that at the steady state the joint velocities converge to zero, indicating that the robot remains at a state of equilibrium, which is the desired state.

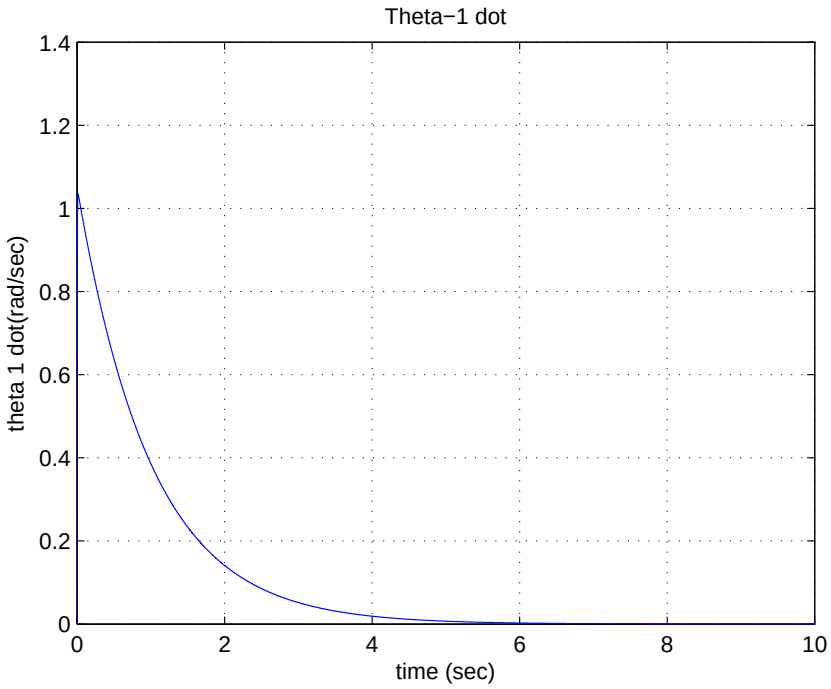


Figure 4.5: Theta 1 dot 'PD control'

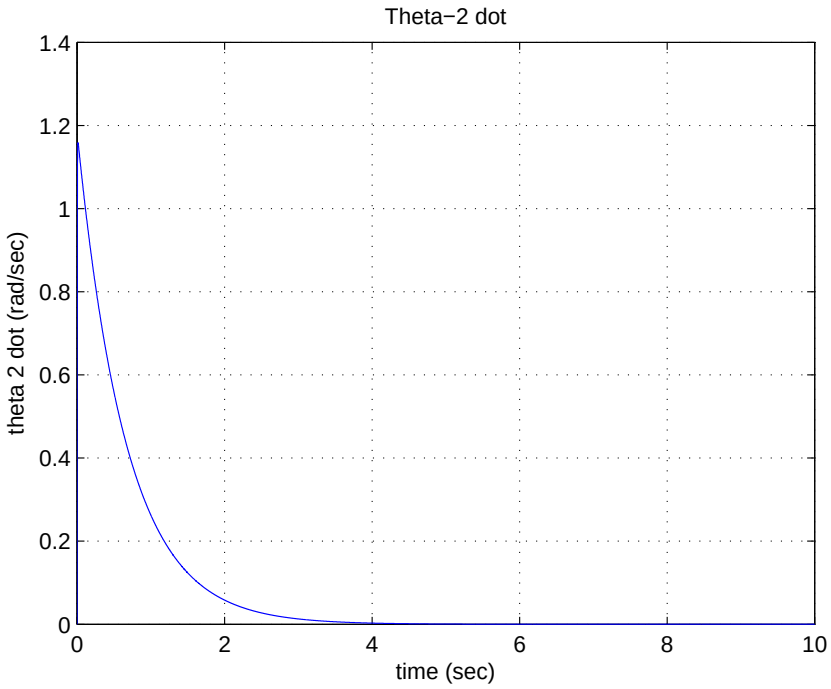


Figure 4.6: Theta 2 dot 'PD control'

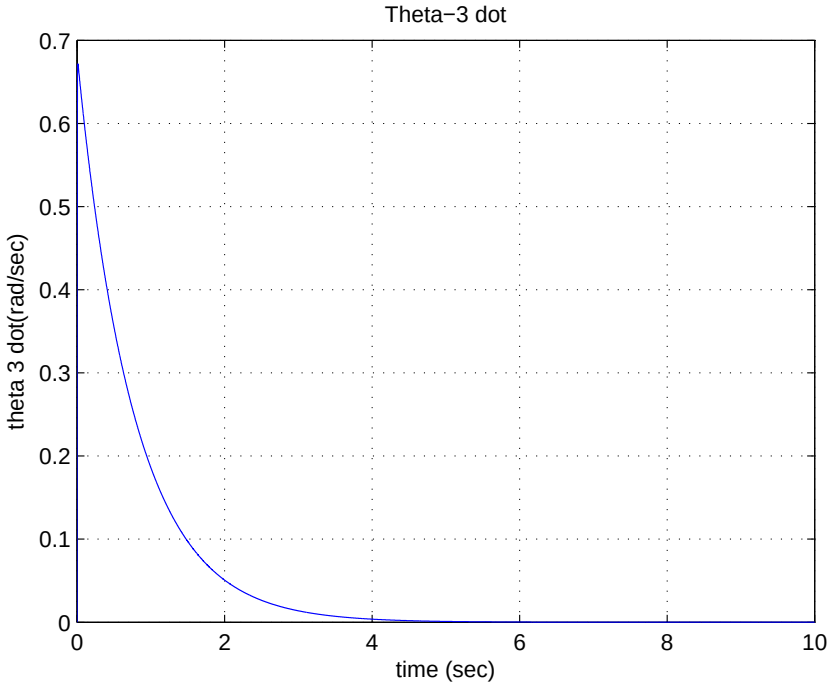


Figure 4.7: Theta 3 dot 'PD control'

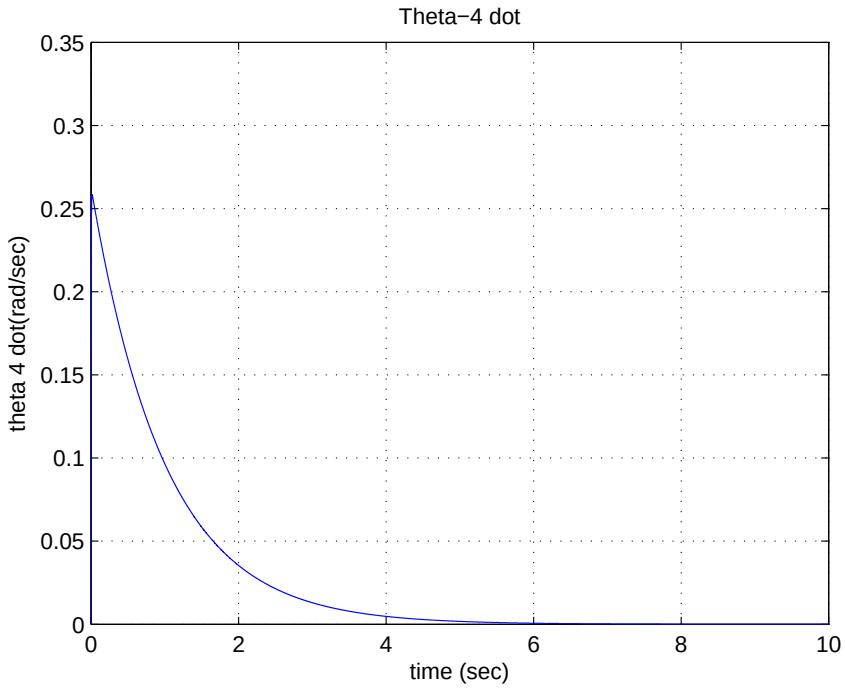


Figure 4.8: Theta 4 dot 'PD control'

4.3.2 PD Control with Gravity Term

In the presence of gravity, following the same analysis, we can show that the dynamics at the equilibrium (in a similar manner to (4.10)) is

$$K_P\tilde{\theta} = g(\theta) \tag{4.12}$$

which can be obtained by direct substitution in (4.9) by considering the gravity term $g(\theta)$. This involves a steady-state error (at the equilibrium point) of

$$\tilde{\theta} = K_P^{-1}g(\theta) \tag{4.13}$$

In this case, depending on the value of the parameters k_P , there is some deviation from the desired position at the equilibrium point.

As for the PD control without gravity term, figures (4.9-4.12) reflect the joint

positions trajectory. Recall that the control law (4.11) is similar to a damper/spring set that controls the joint locations. Because the first link has no effect in the gravitational direction, the controller is able to take that joint to the required final location (4.9). However, this is not the case for the other links and as expected there is an steady state error in the desired position. As figure (4.10) depicts, the position is settled before the desired position. On the other hand, the last two links are settled after the desired position (4.11,4.12), and that is due to their relatively low weights, and the fact that they possess higher centrifugal forces. This offset, nevertheless, must exist so that the joint torque has a steady state value to stabilize the corresponding link's weight.

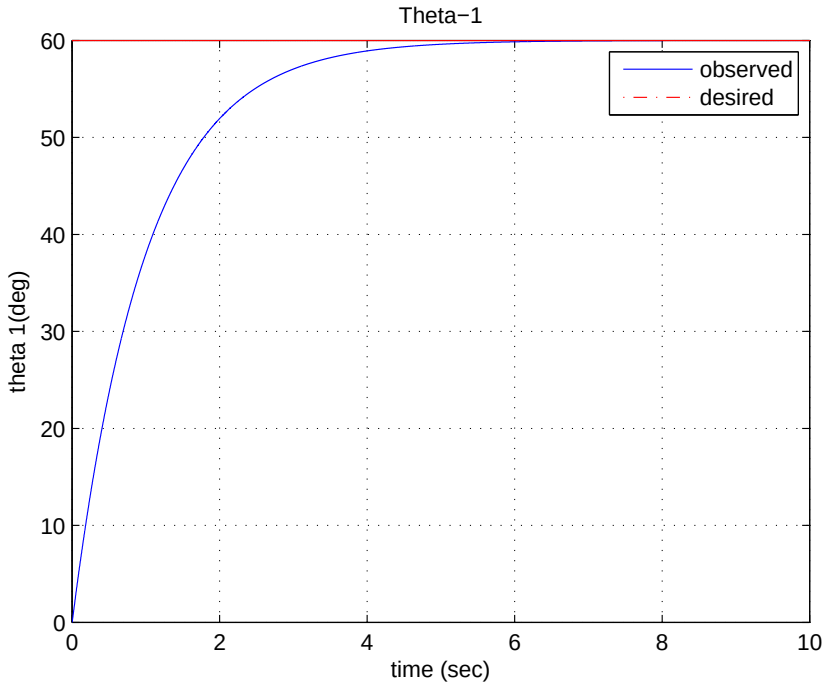


Figure 4.9: Theta 1 Response 'PD control+gravity'

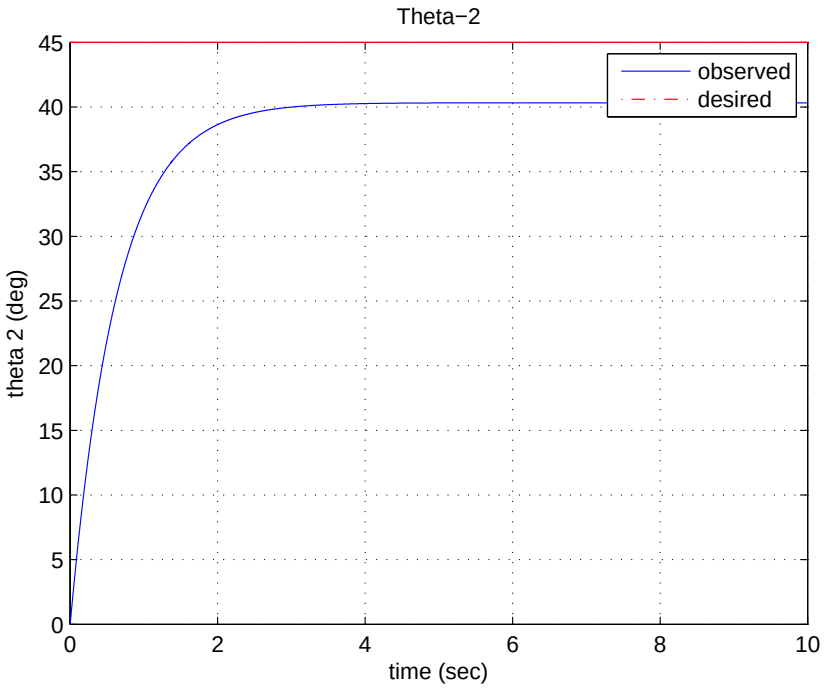


Figure 4.10: Theta 2 Response 'PD control+gravity'

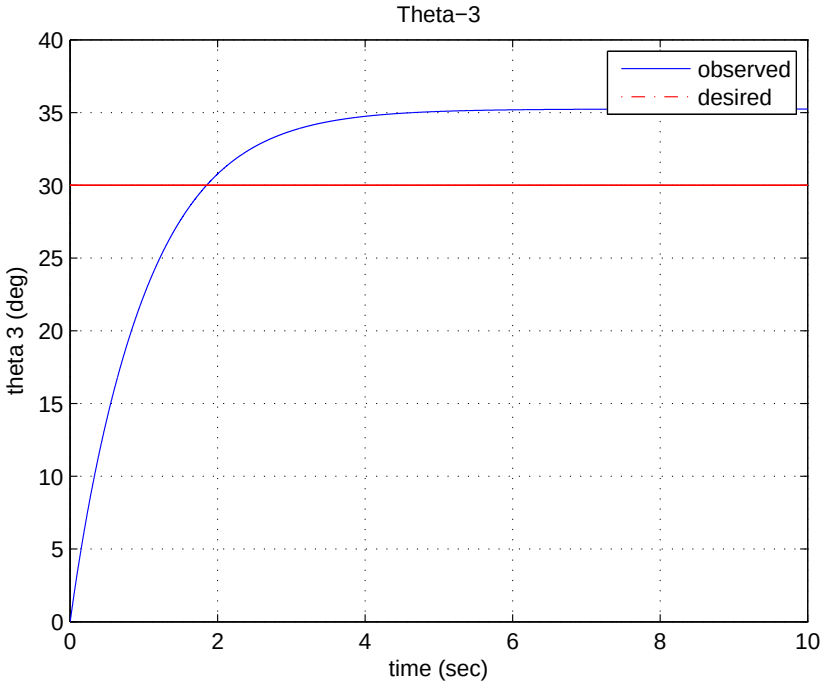


Figure 4.11: Theta 3 Response 'PD control+gravity'

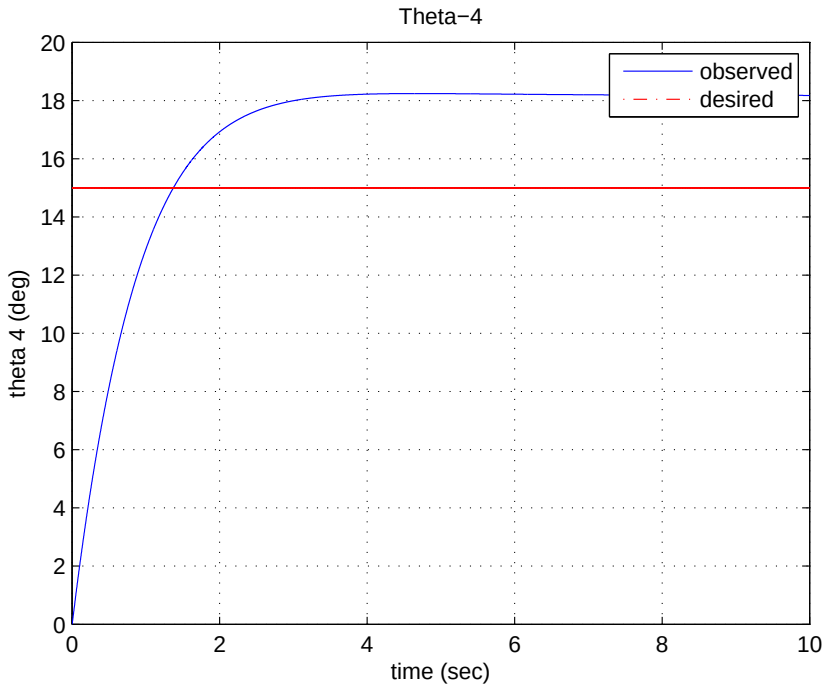


Figure 4.12: Theta 4 Response 'PD control+gravity'

As for the joint velocities, it is clear from figures (4.13-4.16) that at the steady state, the joint velocities disappear, implying that the robotic arm stays at an state of equilibrium.

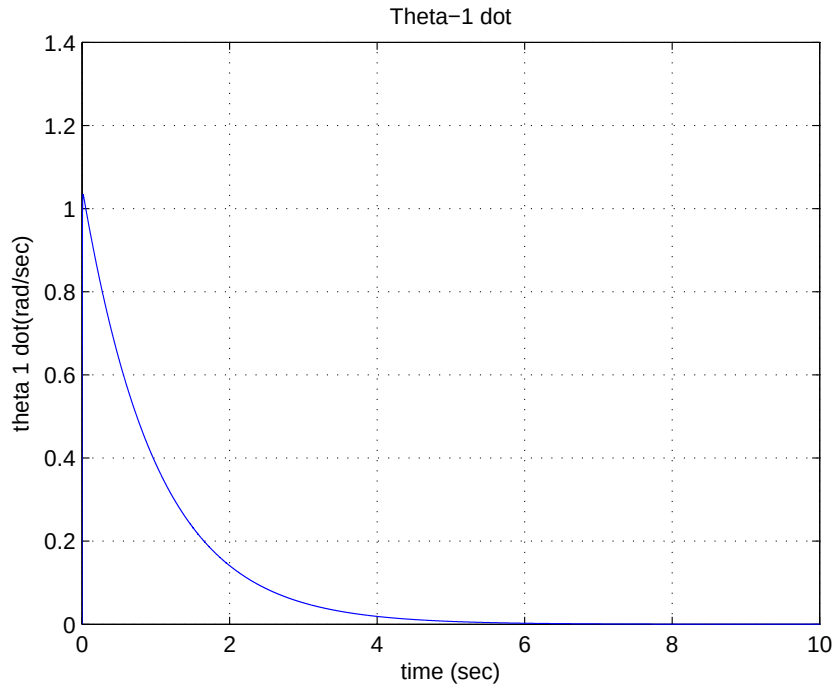


Figure 4.13: Theta 1 dot 'PD control+gravity'

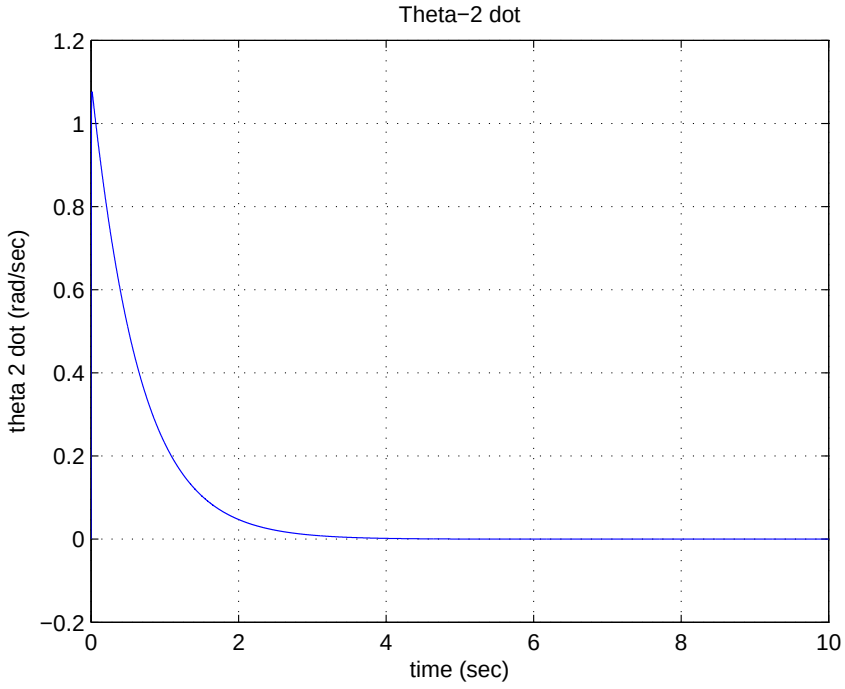


Figure 4.14: Theta 2 dot 'PD control+gravity'

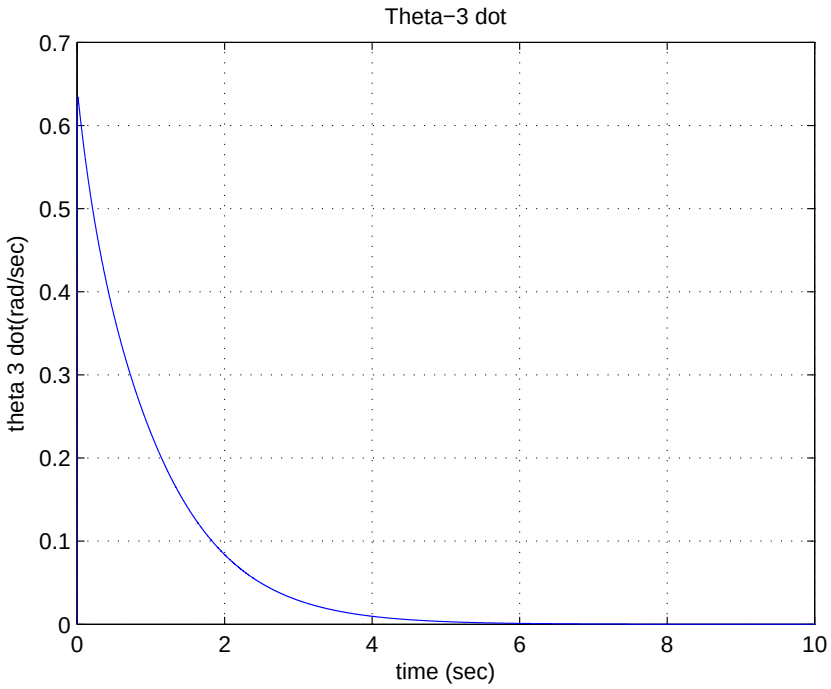


Figure 4.15: Theta 3 dot 'PD control+gravity'

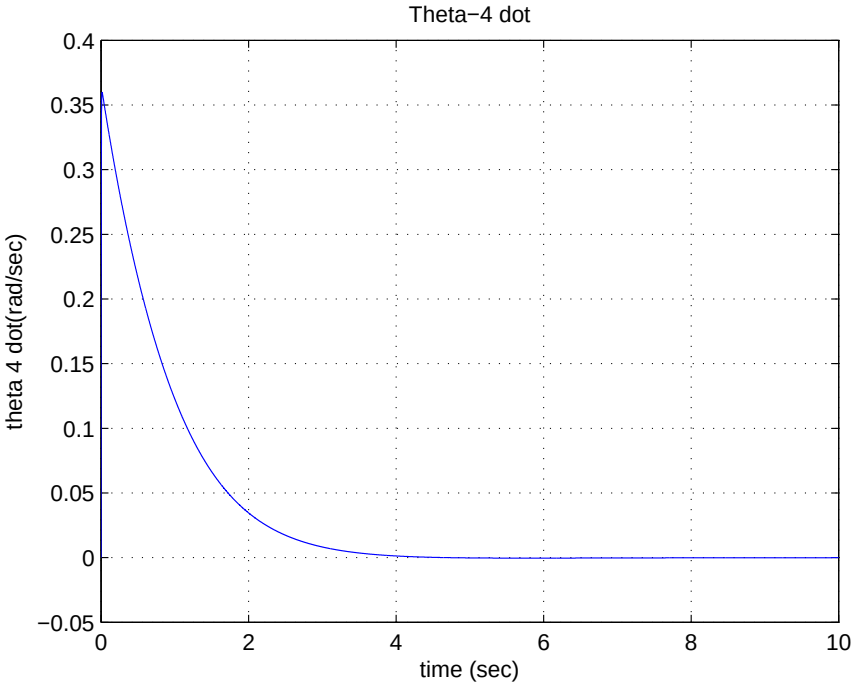


Figure 4.16: Theta 4 dot 'PD control+gravity'

Figures (4.17-4.20) depict the time responses of the joints' torques. It can be shown from figure(4.17) that the torque for the first joint is very small (approaching zero), which confirms the physical fact that since this joint is spinning in the vertical plan, there is no gravitational force acting on it. There is only a small torque exerted at the beginning (0-0.3 sec) that is needed to start the motion and to overcome the friction forces (if there were any). As for the other joints, for rotation in the horizontal plane a constant torque is needed at the steady state to balance the gravity force represented with the weight of the joint. It can be noticed that this torque is relatively small for the last link, due to its light weight.

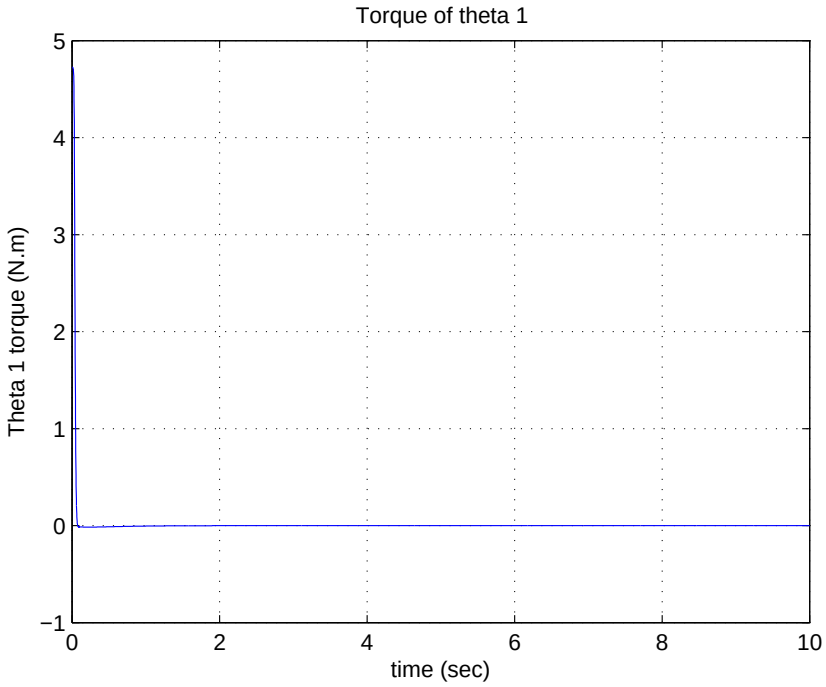


Figure 4.17: Joint 1 Control Signal 'PD+gravity'

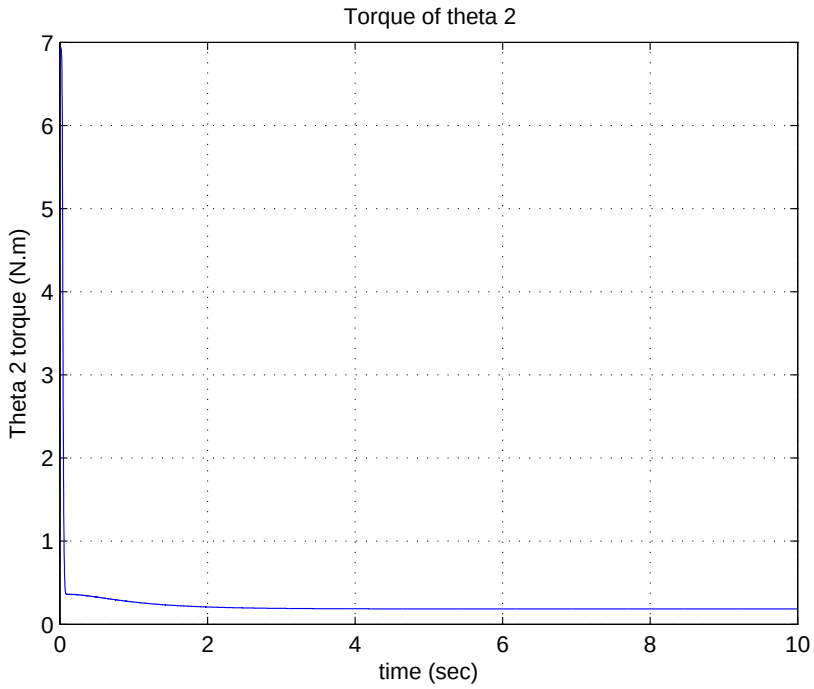


Figure 4.18: Joint 2 Control Signal 'PD+gravity'

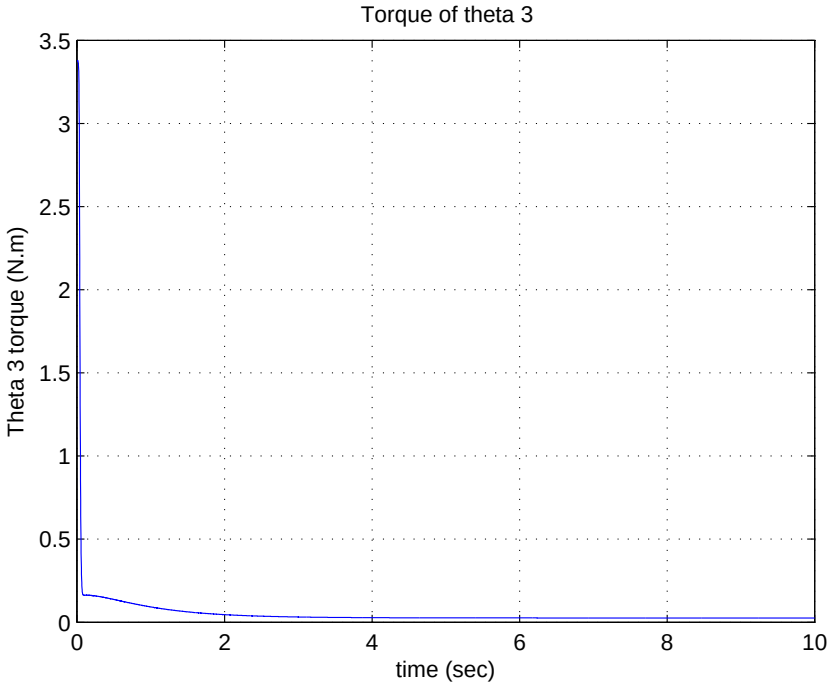


Figure 4.19: Joint 3 Control Signal 'PD+gravity'

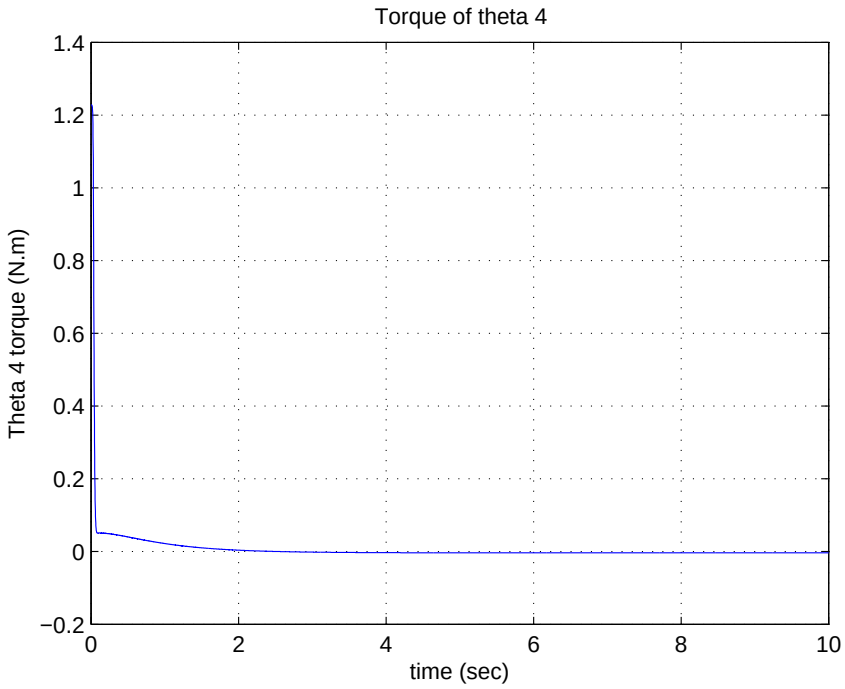


Figure 4.20: Joint 4 Control Signal 'PD+gravity'

4.3.3 PID Control

As we have seen before (section 4.3.1) the PD control is capable of satisfying the position control for the manipulator provided that the gravity term is neglected (i.e $g(\theta) = 0$) in the dynamic model. In this case, the tuning procedure for the PD control is trivial since it is enough to select the gain K_P and K_D to be symmetric and positive definite matrices. In the other case (section 4.3.2) where the dynamic model contains the gravity term ($g(\theta) \neq 0$), then we could not achieve the position control aim by simple means of the PD control law. As seen before (equation 4.13) there is a steady state error. Thus, in order to achieve the required position control aims, the integral component needs to be introduced to the PD control to force the position error to zero. This justifies the necessity to apply the Proportional Integral Derivative (PID) control to the robot manipulator.

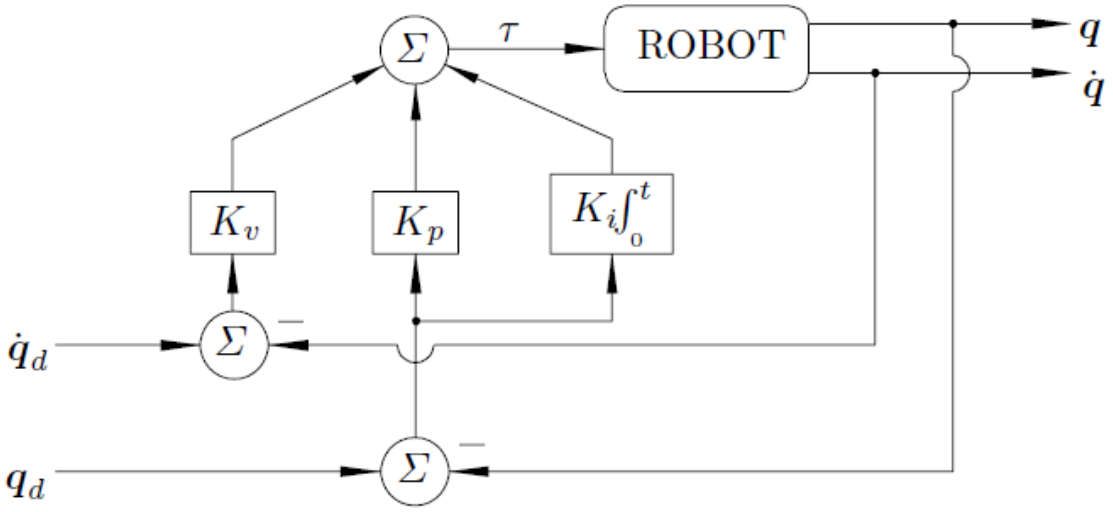


Figure 4.21: Block-diagram: PID control ($q \equiv \theta$)

The PID control is given by:

$$\tau = K_P\tilde{\theta} + K_I \int \tilde{\theta}dt + K_D\dot{\tilde{\theta}} \tag{4.14}$$

where the symmetric positive definite matrices K_P,K_D and K_I are respectively called the proportional, derivative and integral gains, and are chosen properly. Figure (4.21) [43] shows the block-diagram of the PID control for the robot manipulators.

Recently, the PID controllers are used in the control of most of the industrial robotic manipulators. However, unlike the PD control, the tuning methodology for the PID controllers, that is, the methodology to choose suitable parameters K_p, K_v and K_i, is far from trivial. The common approach is to start by selecting random numbers by trial and error and while observing the corresponding response we can get to some extend suitable performance. But, this procedure is not effective and time consuming especially when the degree of freedom is high as will be presented shortly (4.3.4).

Recall that the robot actuators are ideal sources of forces and torques. Under this

assumption, the dynamic model of an n-DOF robot is given by (4.22):

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \tag{4.15}$$

The objective here is to introduce a PID controller given by the formula (4.14), to the robot model given by equation (4.15). The integral action of the PID control law (4.14) introduces an additional state variable that is denoted by ξ, and whose time derivative is $\dot{\xi} = \tilde{\theta}$. The PID control law may be expressed by the following two equations:

$$\tau = K_P\tilde{\theta} + K_I\xi + K_D\dot{\tilde{\theta}} \tag{4.16}$$

$$\dot{\xi} = \tilde{\theta} \tag{4.17}$$

The equation of the closed-loop is obtained by inserting the control action τ from (4.16) in the dynamic model of the robotic manipulator (4.15), i.e.

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = K_P\tilde{\theta} + K_I\xi + K_D\tilde{\theta} \tag{4.18}$$

$$\dot{\xi} = \tilde{\theta} \tag{4.19}$$

which may be written in terms of the state vector $\begin{bmatrix}\xi^T & \tilde{\theta}^T & \dot{\tilde{\theta}}^T\end{bmatrix}^T$ as

$$\frac{d}{dt}\begin{bmatrix}\xi \\ \tilde{\theta} \\ \dot{\tilde{\theta}}\end{bmatrix} = \begin{bmatrix}\tilde{\theta} \\ \dot{\tilde{\theta}} \\ \ddot{\theta}_d - D(\theta)^{-1}[K_P\tilde{\theta} + K_I\xi + K_D\dot{\tilde{\theta}} - C(\theta, \dot{\theta})\dot{\theta} - g(\theta)]\end{bmatrix} \tag{4.20}$$

At the equilibrium, where $\tilde{\theta} = 0$ and $\dot{\tilde{\theta}} = 0$, which implies $\theta = \theta_d$. Hence, the above equation yields $[\xi^T \quad \tilde{\theta}^T \quad \dot{\tilde{\theta}}^T] = [\xi^* \quad 0^T \quad 0^T]$ where

$$\xi^* = K_I^{-1}[D(\theta_d)\ddot{\theta}_d + C(\theta_d, \dot{\theta}_d)\dot{\theta}_d + g(\theta_d)]$$

$$\frac{d}{dt}\begin{bmatrix}\theta_d \\ \dot{\theta}_d\end{bmatrix} = \begin{bmatrix}\dot{\theta}_d \\ D(\theta_d)^{-1}[\tau - C(\theta, \dot{\theta})\dot{\theta} - g(\theta)]\end{bmatrix} \tag{4.21}$$

The Matlab command 'ode45' is used to solve the system of ordinary differential equations.

4.3.4 Manual Tuning

In what follows, an example of tuning the PID controller by trial and error is shown. Although from figure (4.22 and 4.23) the response seems to be reasonable, however, for θ_3 and θ_4 as figure (4.24 and 4.25) depict the response is relatively poor, in terms of both the overshoot and the settling time. The settling time exceeds 10 sec and the overshoot is around 42 % as the worst case for θ_3, figure (4.24).

Manual tuning parameters

$$\theta_i = \begin{bmatrix}0 \\ 0 \\ 0 \\ 0\end{bmatrix} \qquad \theta^d = \begin{bmatrix}\frac{\pi}{3} \\ \frac{\pi}{4} \\ \frac{\pi}{6} \\ \frac{\pi}{12}\end{bmatrix} \qquad K_I = \begin{bmatrix}180 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 \\ 0 & 0 & 903 & 0 \\ 0 & 0 & 0 & 730\end{bmatrix}$$

$$K_P = \begin{bmatrix}650 & 0 & 0 & 0 \\ 0 & 400 & 0 & 0 \\ 0 & 0 & 800 & 0 \\ 0 & 0 & 0 & 500\end{bmatrix} \qquad K_D = \begin{bmatrix}150 & 0 & 0 & 0 \\ 0 & 160 & 0 & 0 \\ 0 & 0 & 900 & 0 \\ 0 & 0 & 0 & 420\end{bmatrix}$$

These are the parameters which are used for manual tuning simulation.

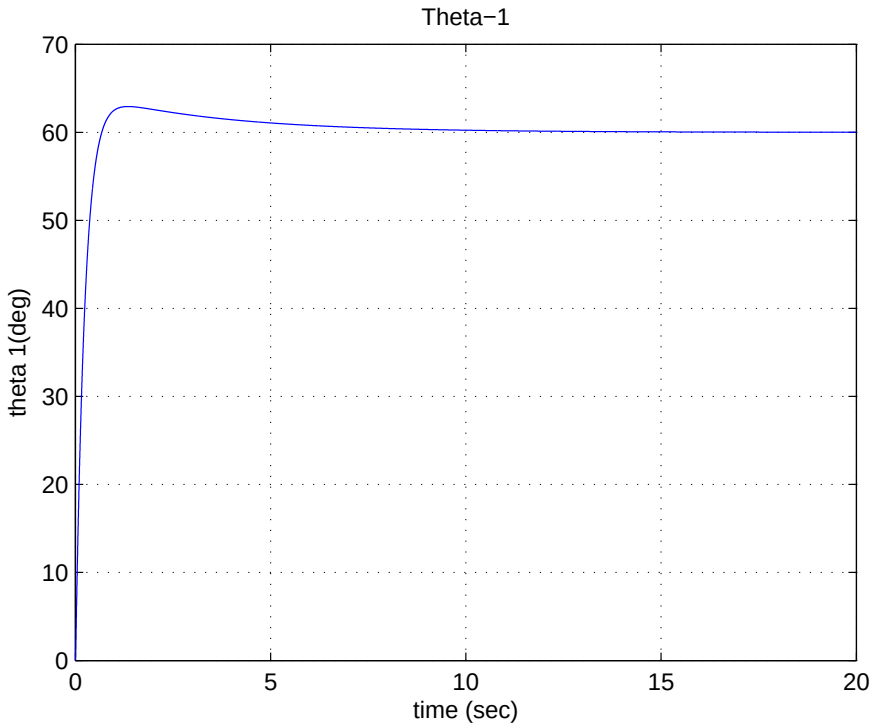


Figure 4.22: Theta 1 Response (manual tuning)

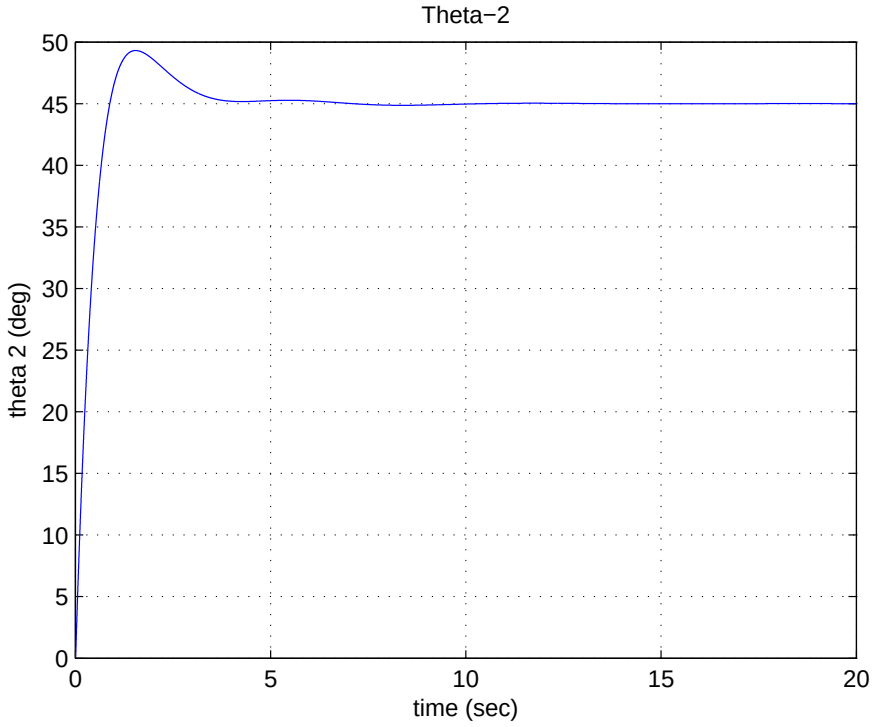


Figure 4.23: Theta 2 Response (manual tuning)

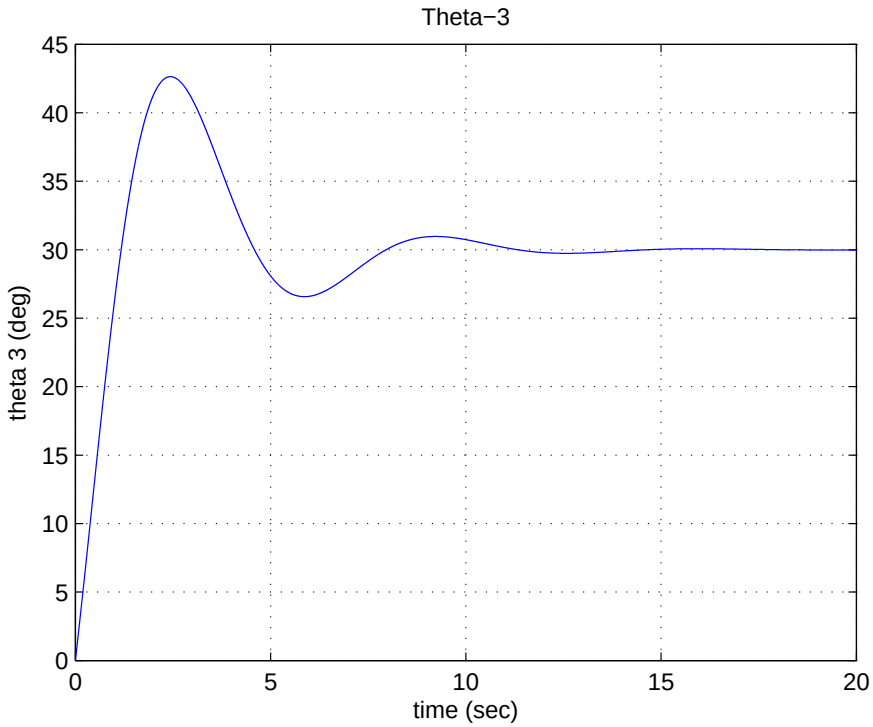


Figure 4.24: Theta 3 Response (manual tuning)

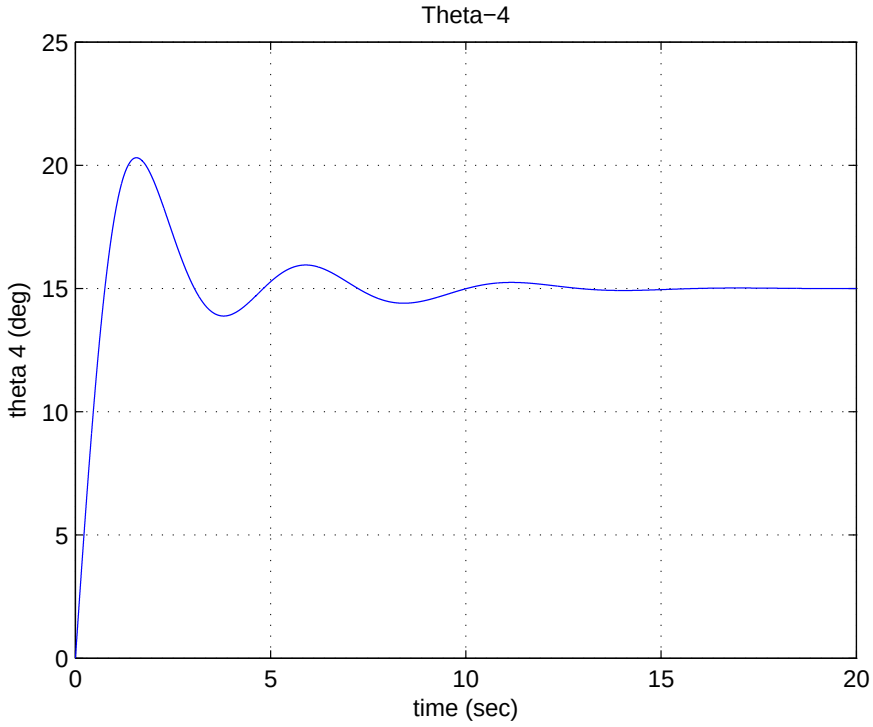


Figure 4.25: Theta 4 Response (manual tuning)

To deal with the high overshoot and settling time, a tuning algorithm needs to be introduced. Because of this, we have used the so-called differential evolution algorithm for the proper tuning of the PID gains.

4.3.5 Differential Evolution

This section introduces the concept of the differential evolution algorithm that we have used for the PID parameters tuning.

Recall the fact that, optimization is a technique used to find the best possible values of decision parameters under a given set of constraints. Differential Evolution (DE) is known to be a very effective global optimizer. Usually optimization techniques pertain to a design problem that will minimize the total cost or maximize the potential reliability or any other known objective [44]. In our case the overall objective is to minimize the overshoot and the settling time. The DE begins with a population of NP candidate solutions which may be denoted by $X_{i,G}, \quad i = 1, ..., NP$, where G represents the generation and i is an index representing the population of that generation. The performance of the DE depends on three main operations: mutation, generation (reproduction) and selection.

The mutation is the core operator of the DE and it is the key operation that makes the DE different from other evolutionary algorithms. The mutation operation of the DE employs vector differentials between the current population individuals for obtaining both the degree and direction of perturbation applied to the individual subject of the mutation operation. At each generation the mutation process starts by choosing three members in the population randomly. [45].

If the preliminary solution is available, the initial population is often produced by adding normally distributed random deviations to that solution. The central idea behind the DE is a new strategy for generating vectors of trial parameters.

The general structure of the Differential Evolution is shown in figure (4.26). For

more details the interested reader may consult the reference [44, 45]. A simplified pseudo code is given as follows:

1 . Randomly initialize the parent population.

2 . Calculate the objective function value $f(X_i)$ for all X_i.

3 . Select three points from population and generate perturbed individual V_i.

4 . Recombine each target vector x_i with perturbed individual generated in step 3 to generate a trial vector U_i.

5 . Check whether each variable of the trial vector is within the range.

6 . Calculate the objective function value for vector U_i.

7 . Choose better of the two (function values at target and trial points).

8 . Check whether convergence criterion is met. If yes then stop, otherwise go to step 3.

In our case the initial population is randomly chosen for the three controller gains $K_P, K_I,$ andK_D, each of them in a range of 0 to 10000. With an original population size of $Np = 20$, a generation size of $Ng = 10$, the cross over factor is $CR = 0.5$ and the mutation factor is $F = 0.5$.

The (overall) objective function is chosen as follows:

$OF = 0.5\times(OF1 + OF2)$ where OF_1 and OF_2 are the individual objective functions for the maximum overshoot and the maximum settling time. Since we want to make those as small as possible, our problem is then an optimization problem to minimize OF. During the coding of this algorithm we asked the program to put the vector of the objective function in ascending order such that the first value in the resulting vector will be corresponding to the best parameter under the prescribed conditions.

The resulting controller parameters are found to be (in the form of positive definite symmetric matrices):

$$K_P = \begin{bmatrix} 9721 & 0 & 0 & 0 \\ 0 & 9721 & 0 & 0 \\ 0 & 0 & 9721 & 0 \\ 0 & 0 & 0 & 9721 \end{bmatrix} \qquad K_D = \begin{bmatrix} 316 & 0 & 0 & 0 \\ 0 & 316 & 0 & 0 \\ 0 & 0 & 316 & 0 \\ 0 & 0 & 0 & 316 \end{bmatrix}$$

$$K_I = \begin{bmatrix} 8354 & 0 & 0 & 0 \\ 0 & 8354 & 0 & 0 \\ 0 & 0 & 8354 & 0 \\ 0 & 0 & 0 & 8354 \end{bmatrix}$$

4.3.6 Control Sequences

For moving the robot arm from one place to another, four scenarios are considered, the shift between each of them being kept as 30 degrees. For the first one, we have started from the home position and have then shifted all the joints by 30 degrees, and have still shifted by another 30 degrees for the second scheme. However, for the generality sake, we have chosen a different initial position for the third scheme from the final position of the second case. We have consequently considered the fourth state by taking the last configuration as the starting point. The obtained results (as shown shortly) reveal that the controller performance is not affected by changing the initial configuration of the manipulator, backing up the fact that the controller can bring the robot between any arbitrary configurations.

The introduction of the integral part into the PD controller cancels the effect of the steady state error and brings the joints the to the desired position in a settling time

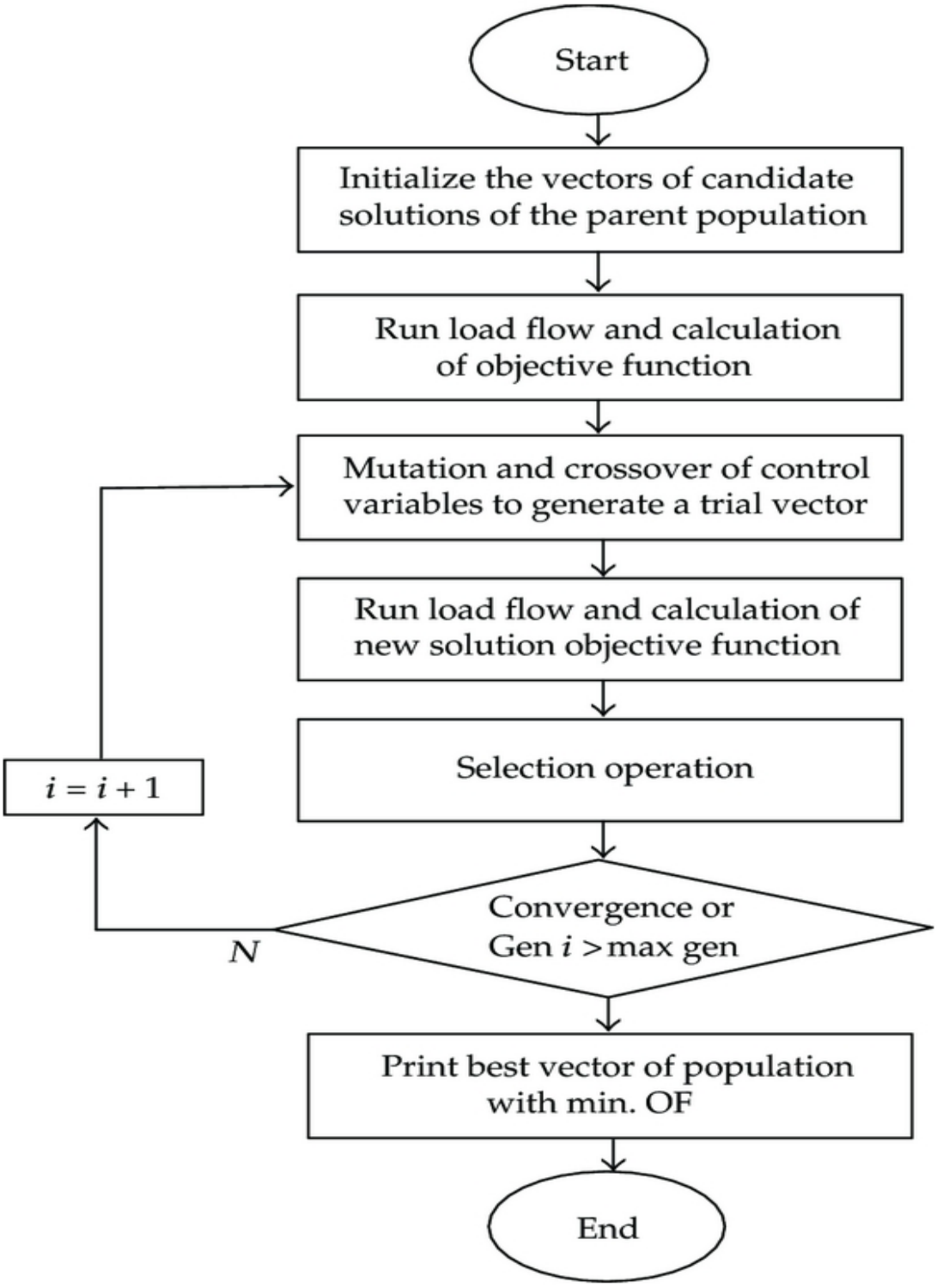


Figure 4.26: Differential Evolution Algorithm

of almost 1 sec with a maximum overshoot of less than 3.4 %. However, we are not claiming the optimality in the performance obtained, but, compared with those results obtained using the manual tuning (4.3.4); the obtained results using DE are more reasonable.

The first Scenario

The initial and final positions (θ and θ^d) are taken as follows:

$$\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad and \quad \theta^d = \begin{bmatrix} \frac{\pi}{6} \\ \frac{\pi}{6} \\ \frac{\pi}{6} \\ \frac{\pi}{6} \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \\ 30 \\ 30 \end{bmatrix}$$

The obtained results are shown in figure (4.27-4.30).

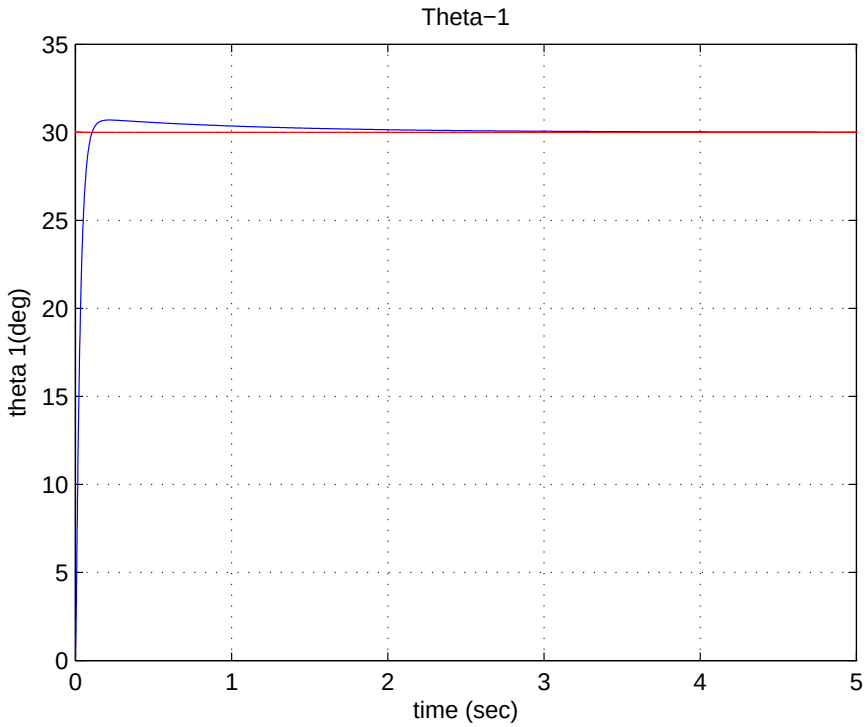


Figure 4.27: Theta 1 Response 'PID control' case 1

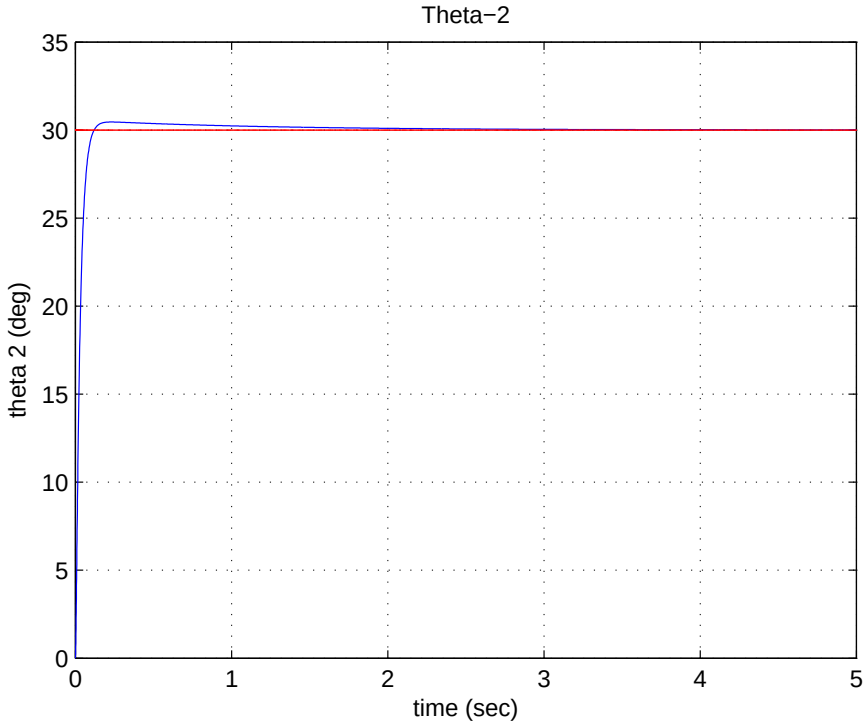


Figure 4.28: Theta 2 Response 'PID control' case 1

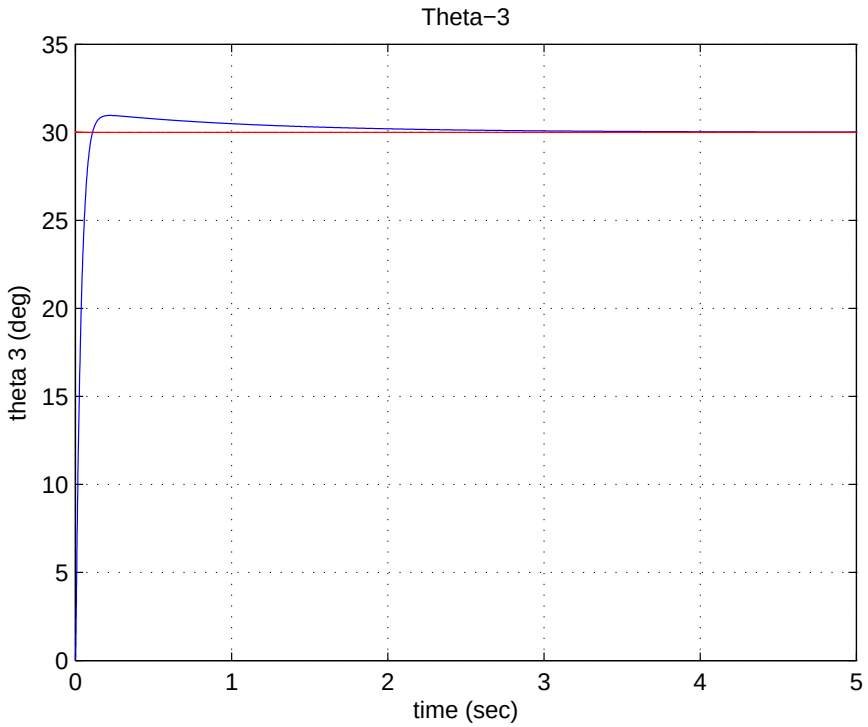


Figure 4.29: Theta 3 Response 'PID control' case 1

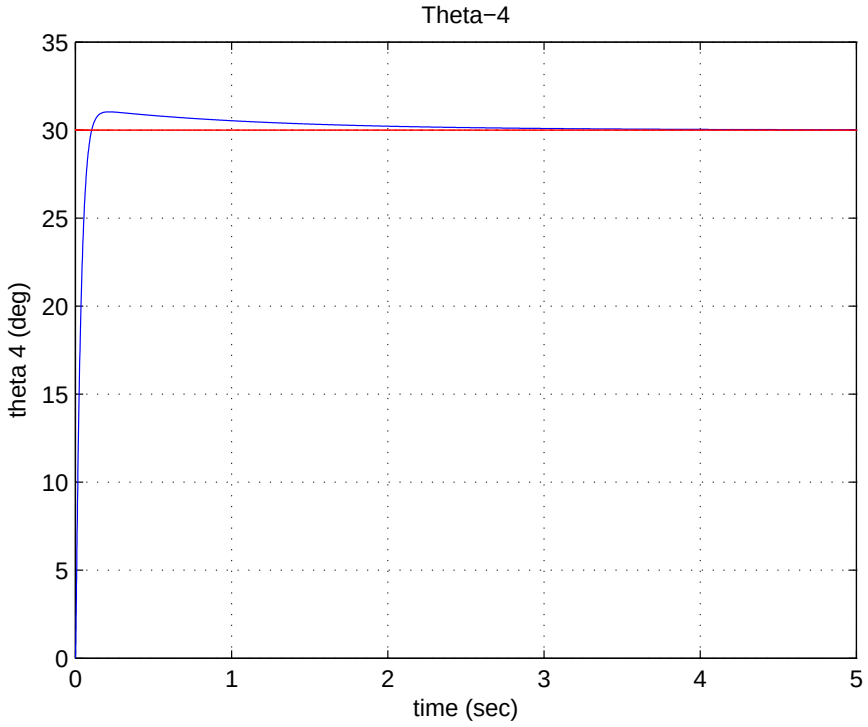


Figure 4.30: Theta 4 Response 'PID control' case 1

The Second Scenario

The initial and final positions (θ and θ^d) are taken as follows:

$$\theta = \begin{bmatrix} \pi/6 \\ \pi/6 \\ \pi/6 \\ \pi/6 \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \\ 30 \\ 30 \end{bmatrix} \quad and \quad \theta^d = \begin{bmatrix} \pi/3 \\ \pi/3 \\ \pi/3 \\ \pi/3 \end{bmatrix} = \begin{bmatrix} 60 \\ 60 \\ 60 \\ 60 \end{bmatrix}$$

The obtained results are shown in figures (4.31,4.32,4.33 and 4.34)

The Third Scenario

The initial and final positions (θ and θ^d) are taken as follows:

$$\theta = \begin{bmatrix} \pi/2 \\ \pi/3 \\ \pi/4 \\ 5\pi/12 \end{bmatrix} = \begin{bmatrix} 90 \\ 60 \\ 45 \\ 75 \end{bmatrix} \quad and \quad \theta^d = \begin{bmatrix} 2\pi/3 \\ \pi/2 \\ 5\pi/12 \\ 7\pi/12 \end{bmatrix} = \begin{bmatrix} 120 \\ 90 \\ 75 \\ 105 \end{bmatrix}$$

The obtained results are shown in figures (4.35-4.38)

The Fourth Scenario

The initial and final positions (θ and θ^d) are taken as follows:

$$\theta = \begin{bmatrix} 2\pi/3 \\ \pi/2 \\ 5\pi/12 \\ 7\pi/12 \end{bmatrix} = \begin{bmatrix} 120 \\ 90 \\ 75 \\ 105 \end{bmatrix} \quad and \quad \theta^d = \begin{bmatrix} 5\pi/6 \\ 2\pi/3 \\ 7\pi/12 \\ 3\pi/4 \end{bmatrix} = \begin{bmatrix} 150 \\ 120 \\ 105 \\ 135 \end{bmatrix}$$

The obtained results are shown in figures (4.39-4.41)

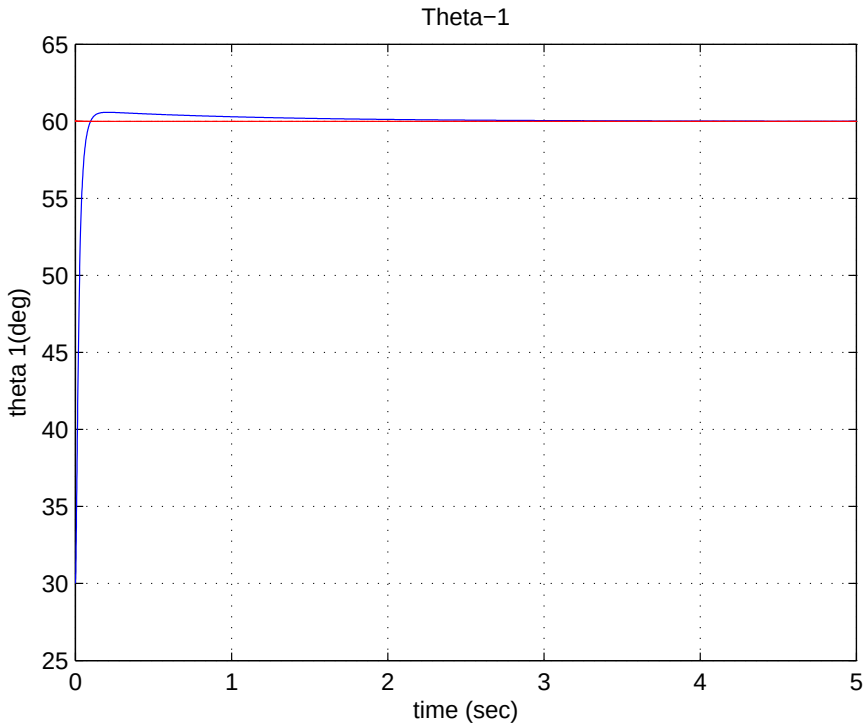


Figure 4.31: Theta 1 Response 'PID control' case 2

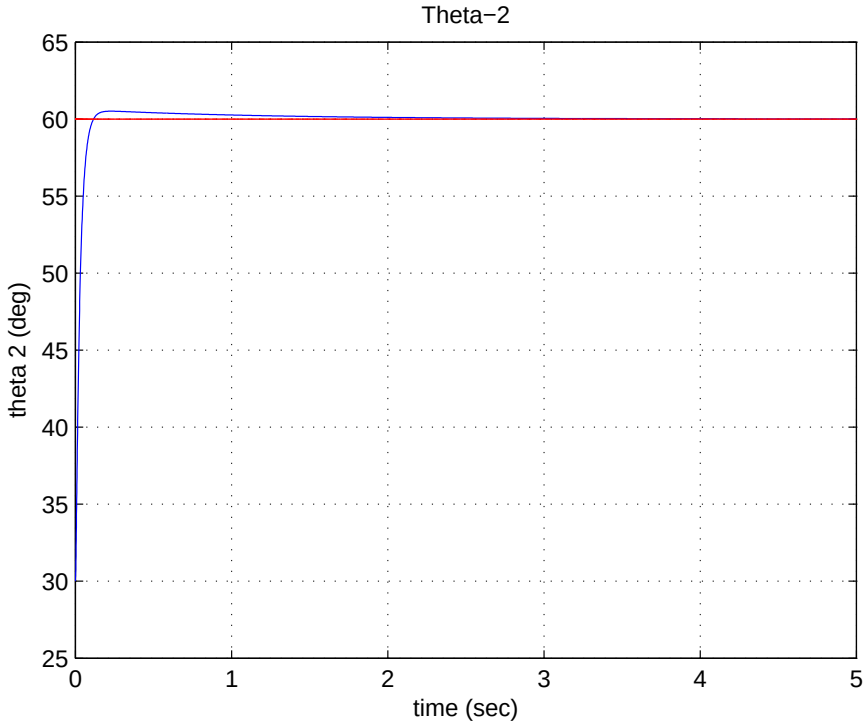


Figure 4.32: Theta 2 Response 'PID control' case 2

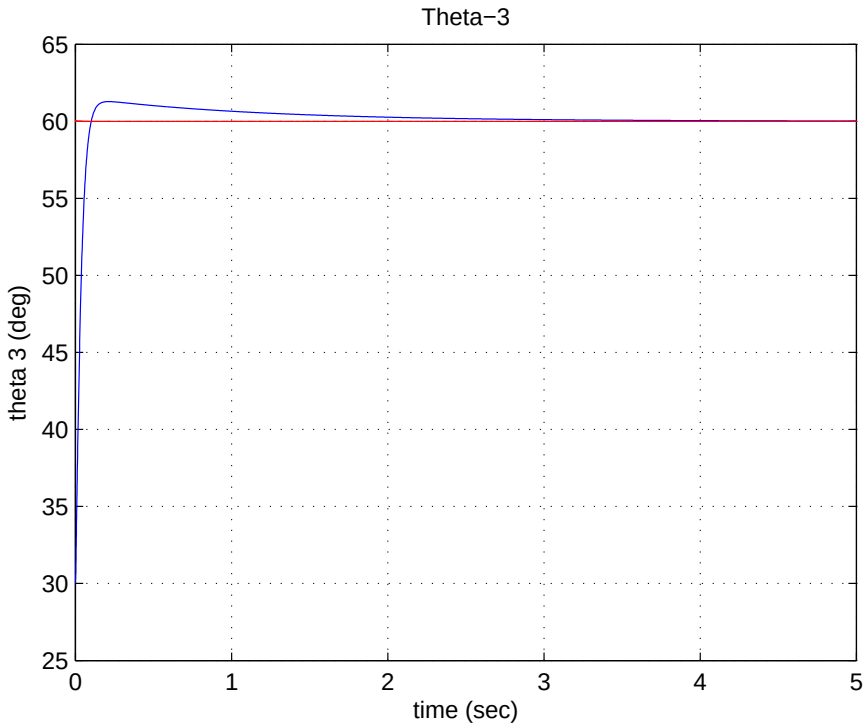


Figure 4.33: Theta 3 Response 'PID control' case 2

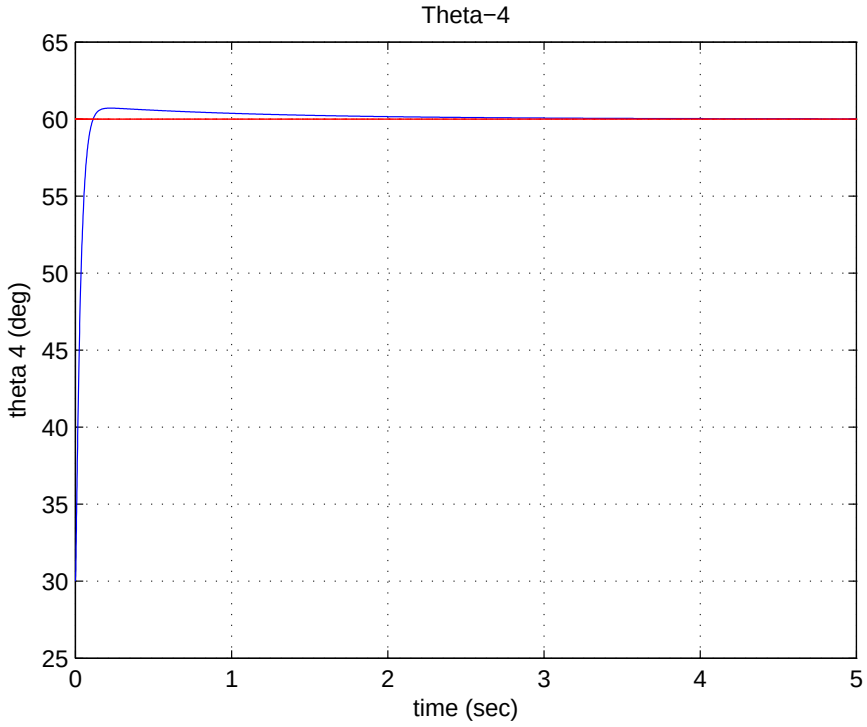


Figure 4.34: Theta 4 Response 'PID control' case 2

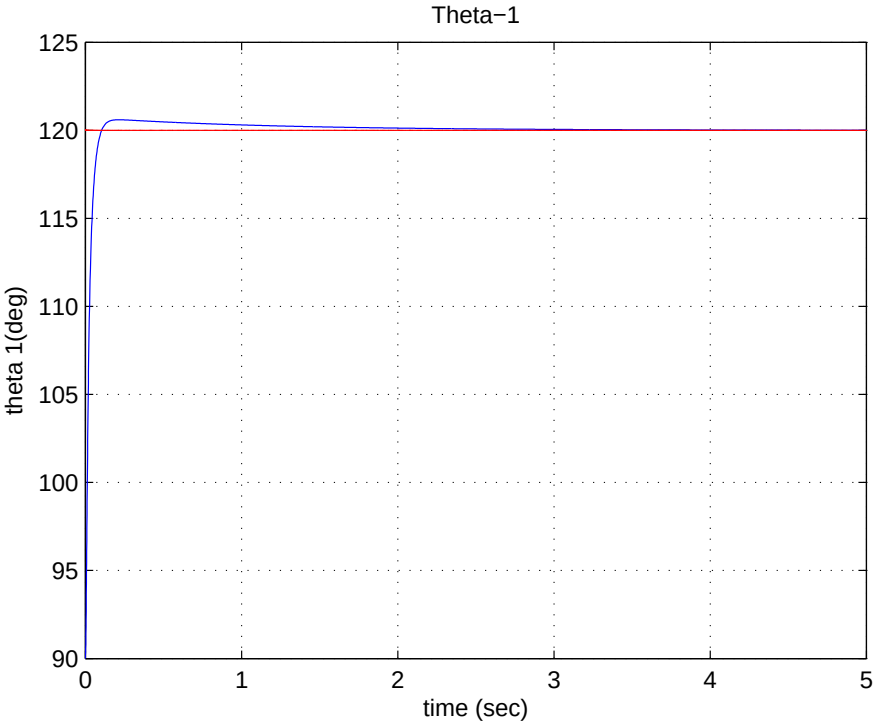


Figure 4.35: Theta 1 Response 'PID control' case 3

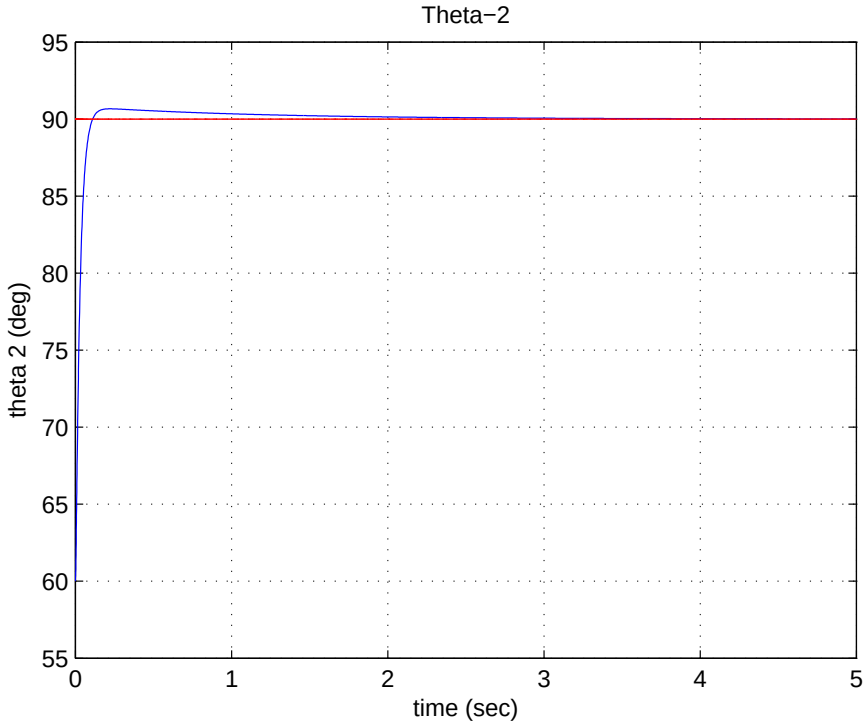


Figure 4.36: Theta 2 Response 'PID control' case 3

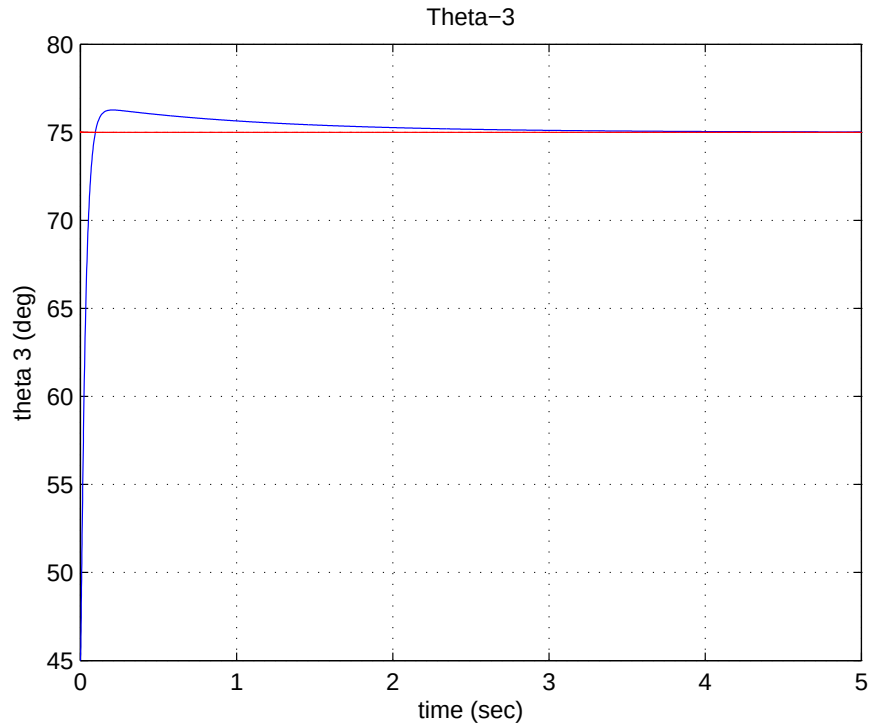


Figure 4.37: Theta 3 Response 'PID control' case 3

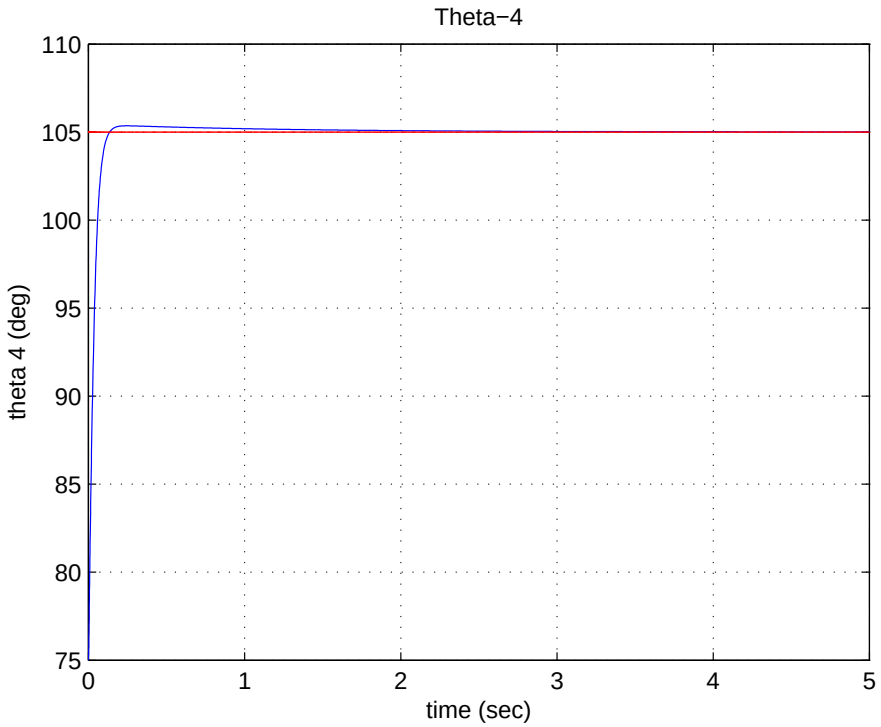


Figure 4.38: Theta 4 Response 'PID control' case 3

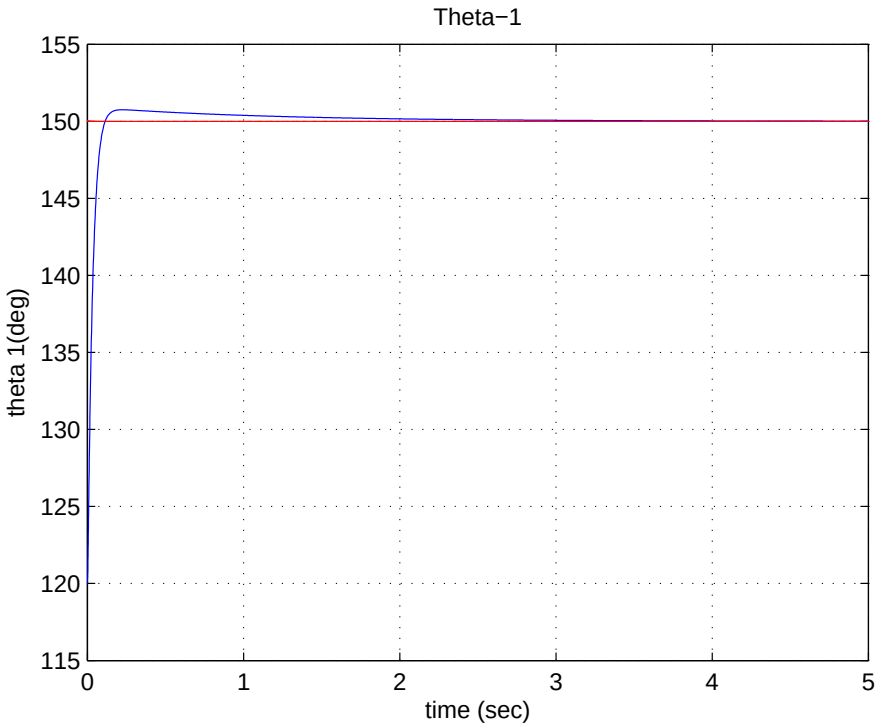


Figure 4.39: Theta 1 Response 'PID control' case 4

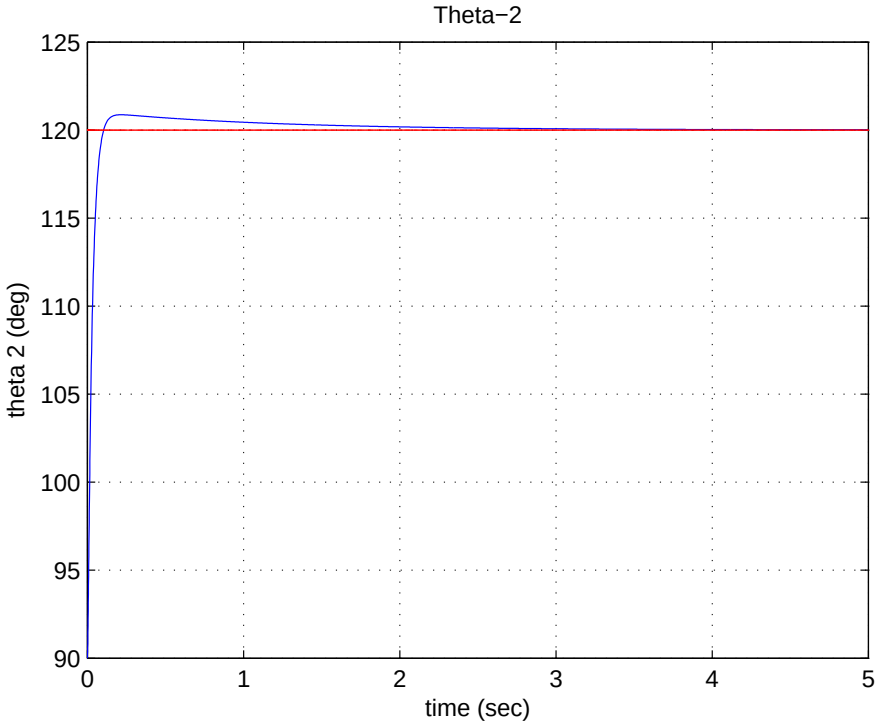


Figure 4.40: Theta 2 Response 'PID control' case 4

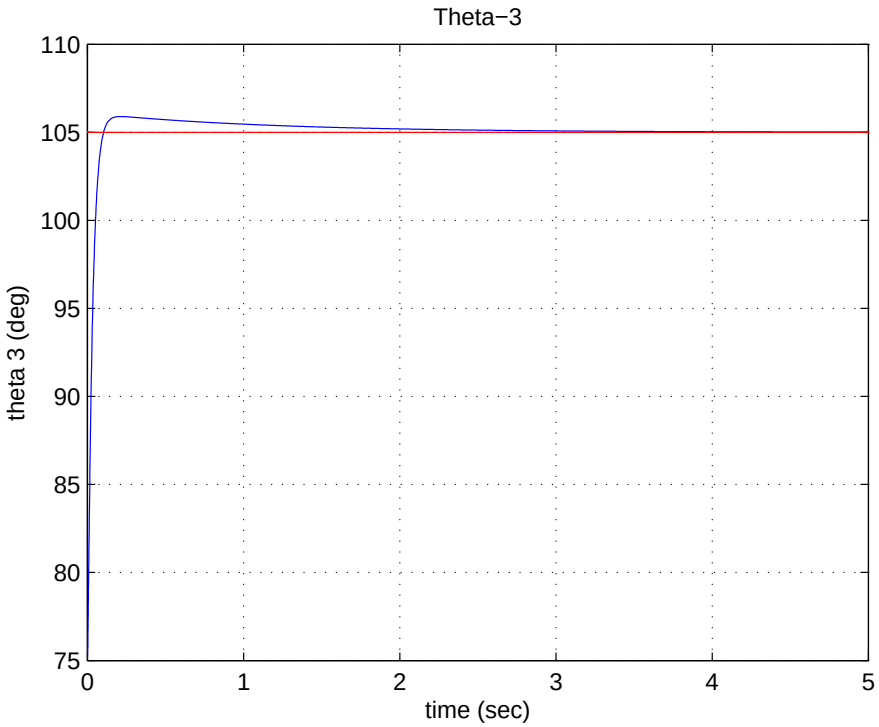


Figure 4.41: Theta 3 Response 'PID control' case 4

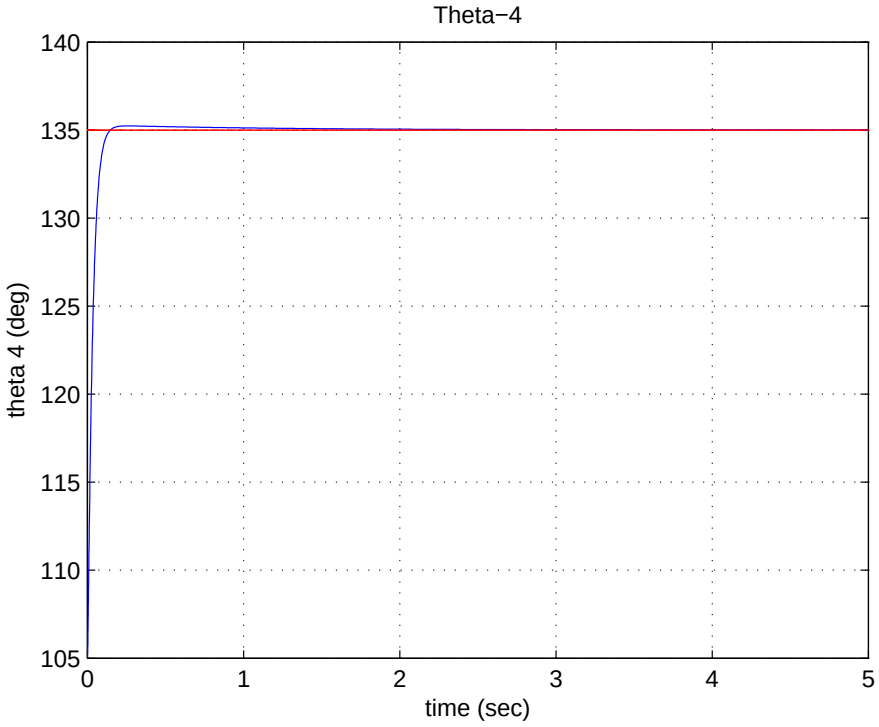


Figure 4.42: Theta 4 Response 'PID control' case 4

4.3.7 Position Plots

For direct visualization, it is very useful to show the different configurations of the robot while explaining the effect of the external forces on the control specification, especially, the effect of the gravitational forces. Using the robotics toolbox for MATLAB [33, 46], as introduced before, we can produce figures that display the robot manipulator in each scenario.

4.3.8 Results and Analysis of PID Control

This section presents the obtained results using the PID control for the 4-DOF robotic manipulator. Since the robot is targeted to move from one place to another for picking and placing purposes, different sets of initial and final positions are investigated; the obtained results are analyzed and discussed. Moreover, since the is no change at the end, the time span is reduced to 5 seconds and a comprehensive summary of the obtain results is shown in table (4.1). By direct inspection of that table (i.e. table 4.1), it is clear that the biggest overshoot occurring is 3.3943 % in the first case corresponding to the fourth joint angle. The physical interpretation for this is that the fourth link is furthest from the origin, meaning that effects of centrifugal and Coriolis forces are bigger as compared with the other joint angles. So, as a result, the larger the distance from the origin, the bigger the effect of the gravity on the controlled behavior. A similar trend is seen earlier with the PD control. Moreover, from table (4.1), we can see for both the properties of overshoot and the settling time that for the planar three links (link 2, 3, and 4) these properties are proportional to the distance from the origin in ascending manner with the exception of the first link which is not affected by the gravity at all. However, for the rise time, the issue is a bit different. We can see that the highest rise time in this particular case is 0.0638 sec corresponding to θ_3 (link 3), while it is 0.0606 sec for (link 4). This could be interpreted as follows.

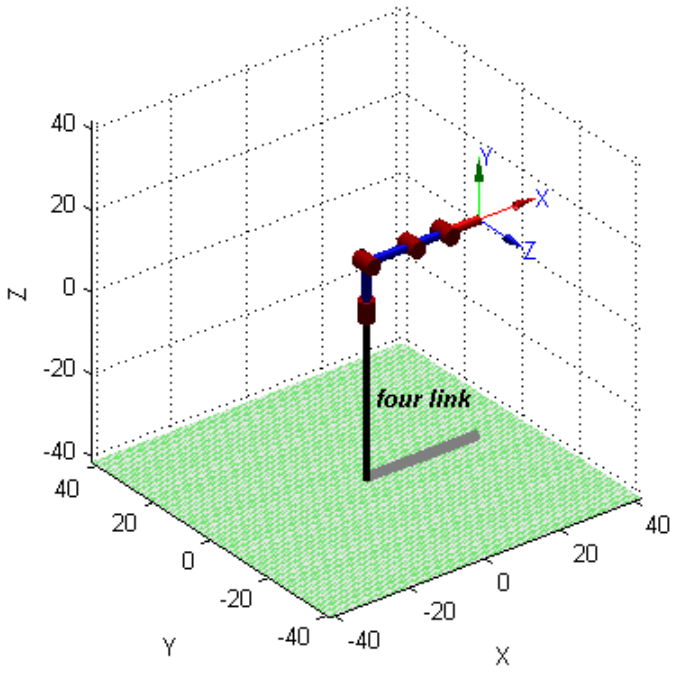


Figure 4.43: Home/ initial position, case 1

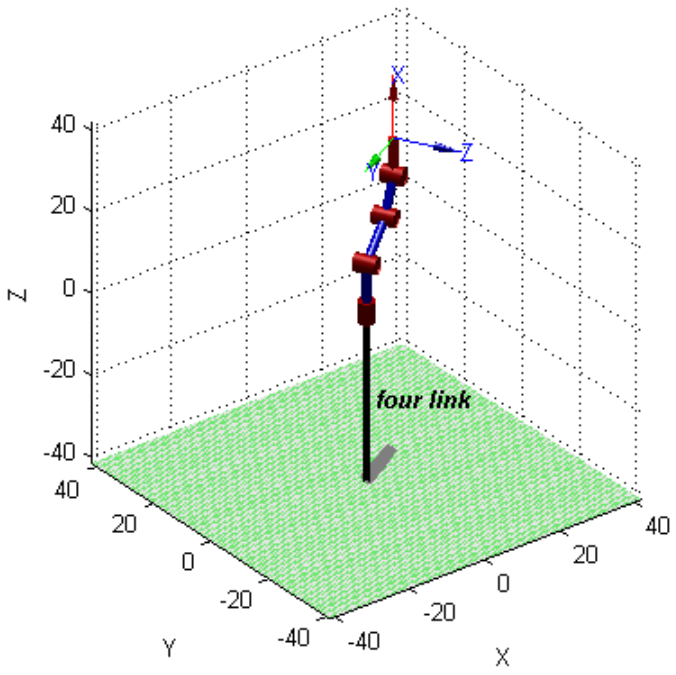


Figure 4.44: Desired position, case 1

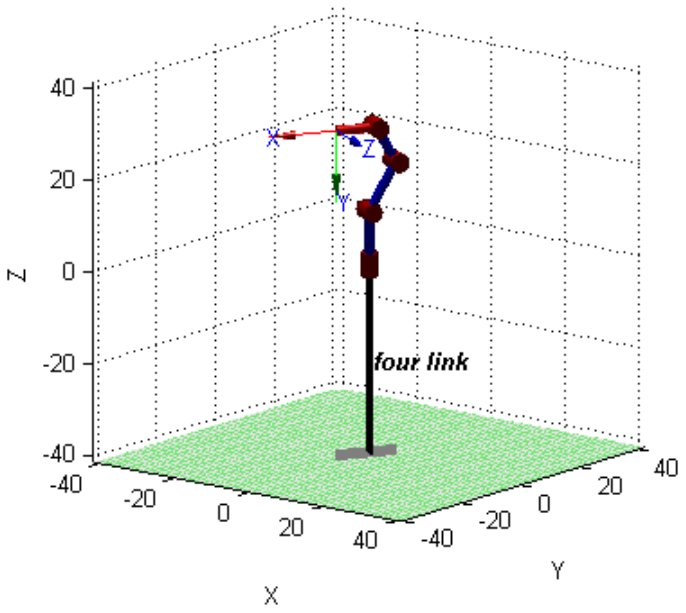


Figure 4.45: Desired position, case 2

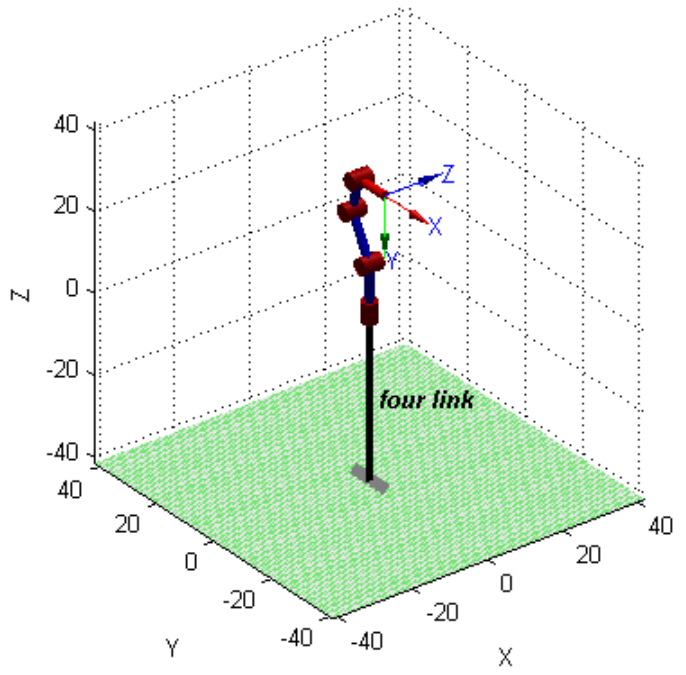


Figure 4.46: Initial position, case 3

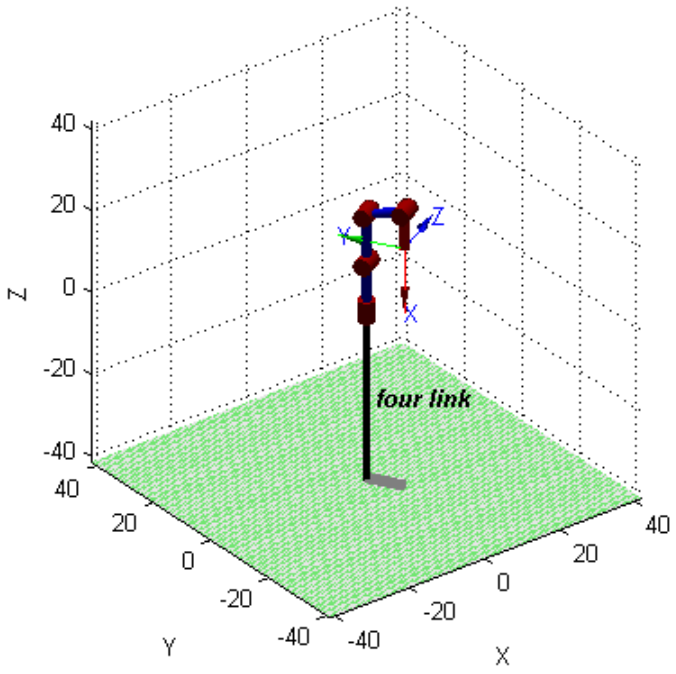


Figure 4.47: Desired position, case 3

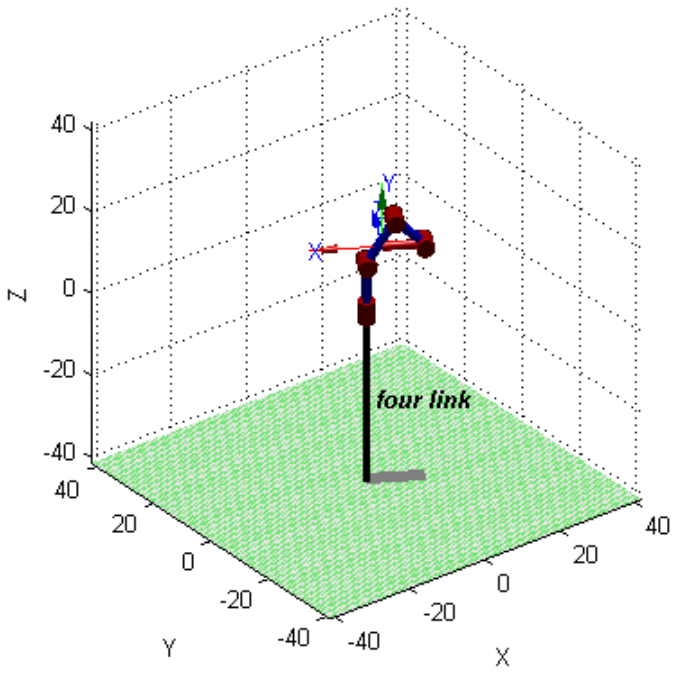


Figure 4.48: Desired position, case 4

When the links are moving to a 30 degree position, the fourth link is far away from the horizontal and it tends to reach the stable position of 90 degrees figure (4.44) rather than staying at the home position figure (2.4), which would need a higher shift back. That is why it goes faster to catch the upright stable position. It is worthy to mention that the end effector orientation is independent from the base rotation of link 1, thus its main contribution is in the position of the end effector in the global frame. Revisiting table (4.1), we can see that for case II, the maximum overshoot observed is 2.1052 % for θ_3, which is greater than that of θ_4 (just around 1.1633 %). This can be justified in a similar manner as before. In other words, after the first motion, the link 4 is in the upright stable position figure (4.44). When the second journey starts, the link 3 exceeds the upright position figure (4.45), then the link 4 moves down in the gravity direction. This means that the gravity has a positive effect on the overshot in this case and it helps the link 4 to move easily reducing the overshoot, the steady state error and consequently the settling time. It is clear from table (4.1) that the settling time of θ_4 is less than that of θ_3, and rise times of θ_3 and θ_4 are almost identical. Considering the third case, the highest overshoot observed is 1.6660 % corresponding to the joint variable of link 3, which can be explained as follows. Considering the initial configuration of case 3 figure (4.46), the link 4 is 15 degree above the horizontal. When all the joint angles are shifted by 30 degrees where the link 2 becomes aligned upward with the link 1, as figure (4.47) depicts, the link 3 is close to the horizontal (around 15 degree above) and the fourth link is hanging downward in almost the vertical direction (making 105 degrees with respect to the previous link). This means that the gravity force has a higher effect on link 3 than on link 4, because the center of gravity of link 3 in this case is well above that of link 4. This observation reinstates the relation between the gravity force and the overshoot in a proportional manner. In other words, when the link is reacting/balancing a high gravity force (opposite to the gravitational direction) it generates a big overshoot on

the corresponding joint variable. On the contrary, when the link is moving in the direction of gravity, the related overshoot is expected to be low. Similar argument is applicable for the settling time that is related in turn to the overshoot. It is interesting to note that the treatment of case (III) goes along with that of case (II). As for the rise time, it can be seen to be 0.0670 sec for θ_4, which is greatest for this case, and this is because of the high centrifugal force acting on link 4. For the fourth case, we can see again that the highest overshoot is associated with the joint variable θ_3 and that is because the third link in this case is oriented by 45 deg (which is highly unstable position due to effect fo gravity) and is affected by the weight of the next link. As stated before, the further from the origin the link is, the higher the overshoot is expected to be. But 0.8396 % overshoot for the link 3 is even higher than 0.1723 % overshoot of link 4 although the link 4 is the furthest link. This can be argued here such that, the center of gravity of link 3 is higher from the datum than that of link 4, making the gravity effect on link 3 higher for the overshoot and settling time. The largest rise time monitored in this case is 0.0662 sec corresponding to the join variable θ_4, and this is also due to the high centrifugal force on the link further away from the origin.

Table 4.1: Results Summary

Case	θ_i	θ_f	Overshoot	Settling time	Rise time
	0	30	2.2940	0.4007	0.0563
I	0	30	1.5022	0.0976	0.0583
	0	30	3.1471	0.7505	0.0638
	0	30	3.3943	0.8327	0.0606
	30	60	0.9611	0.0790	0.0449
II	30	60	0.8398	0.0954	0.0570
	30	60	2.1052	1.0619	0.0616
	30	60	1.1633	0.4252	0.0618
	90	120	0.4925	0.0859	0.0495
III	60	90	0.7285	0.3496	0.0573
	45	75	1.6660	1.0508	0.0568
	75	105	0.3322	0.1094	0.0670
	120	150	0.4922	0.4856	0.0625
IV	90	120	0.7145	0.6395	0.0568
	75	105	0.8396	0.6768	0.0581
	105	135	0.1723	0.1121	0.0662

The obtained results are further summarized in table (4.2), where only the maximum values for each property in the specified case are included. It is clear from this table the maximum value for the settling time is around 1.0619 sec which takes place in case 2, while the maximum overshoot observed is 3.3943 % that occurs in the first case, and the maximum rise time is 0.0670 sec for case 3. The time span used here is 5 sec.

Table 4.2: Results Summary (maximum values)

	case 1	case 2	case 3	case 4
Settling time Max (sec)	0.8327	1.0619	1.0508	0.6768
Overshoot Max(%)	3.3943	2.1052	1.6660	0.8396
Rise time (sec)	0.0638	0.0618	0.0670	0.0662

For the joint velocities, it is clear from figures (4.49-4.52) that, at the steady state the joint velocities disappear (reach zero) in around 0.2 sec, indicating that the robot stays at a state of equilibrium.

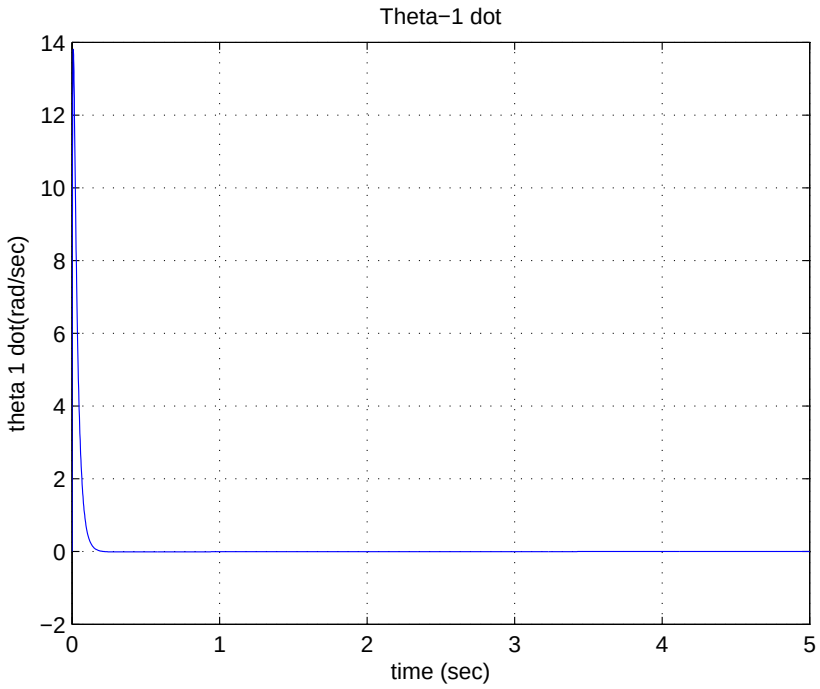


Figure 4.49: Theta 1 dot 'PID control' case 1

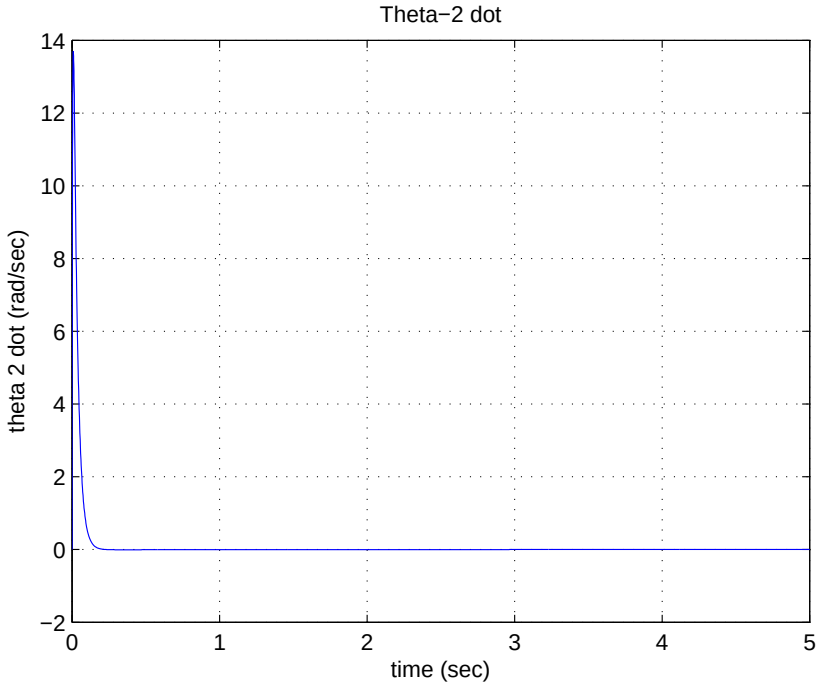


Figure 4.50: Theta 2 dot 'PID control' case 1

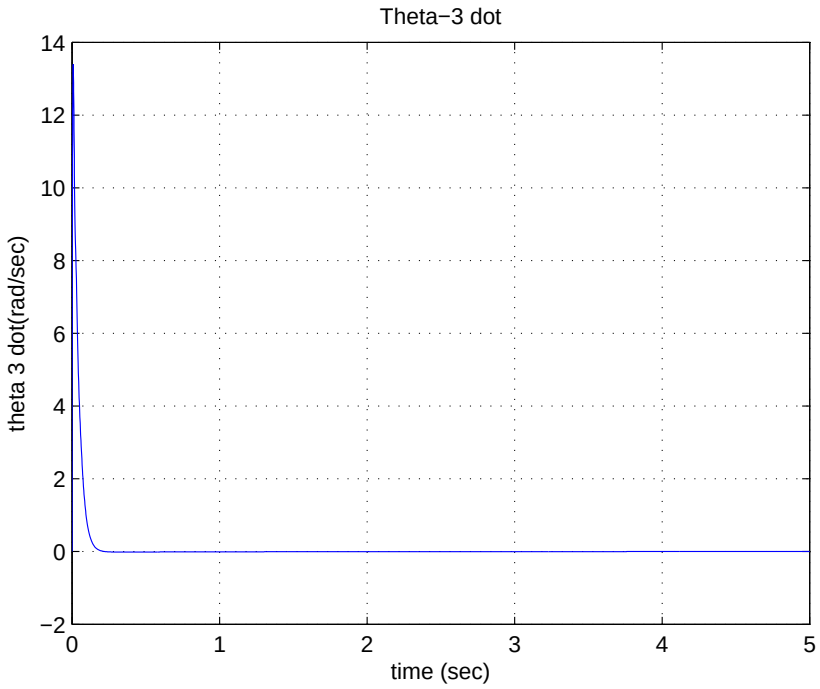


Figure 4.51: Theta 3 dot 'PID control' case 1

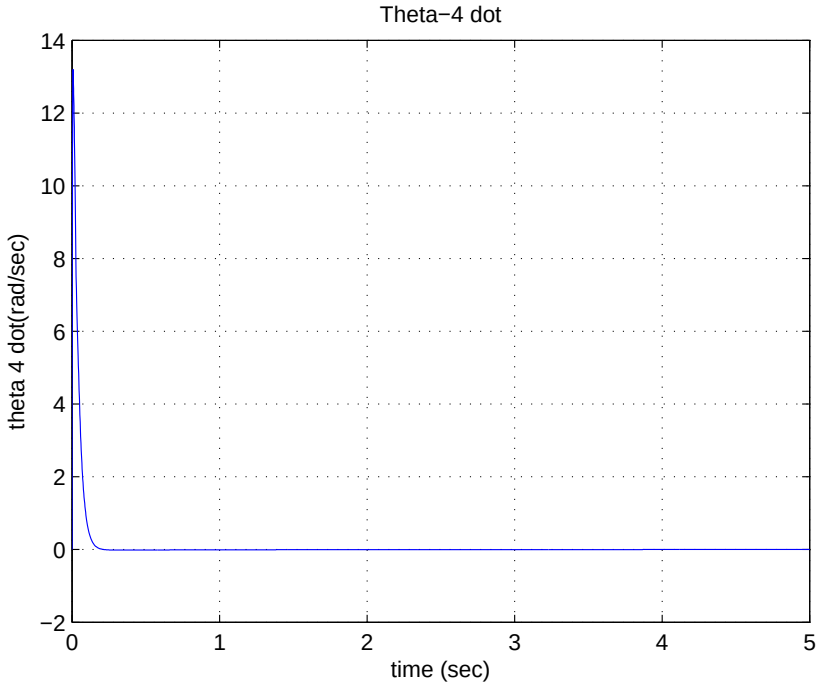


Figure 4.52: Theta 4 dot 'PID control' case 1

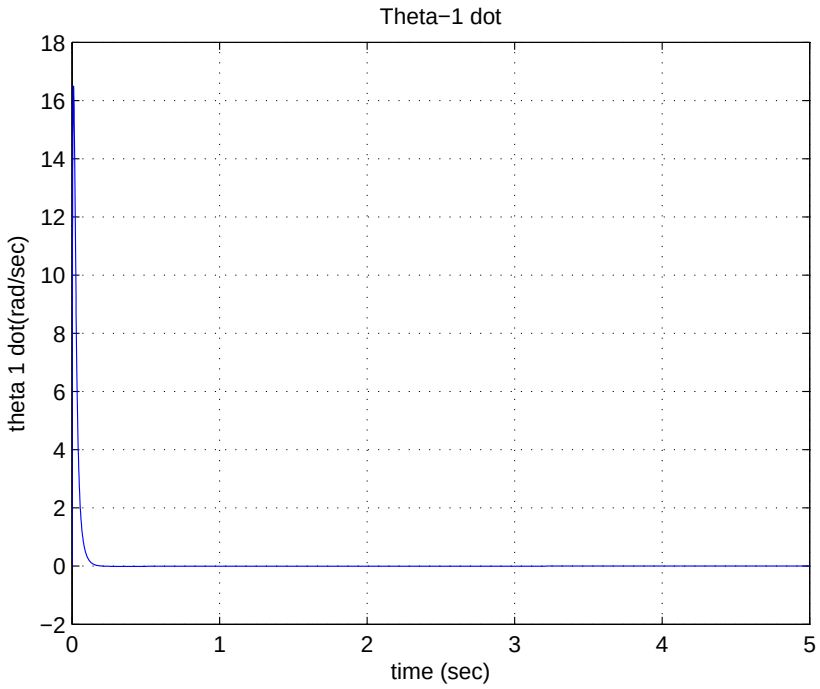


Figure 4.53: Theta 1 dot 'PID control' case 2

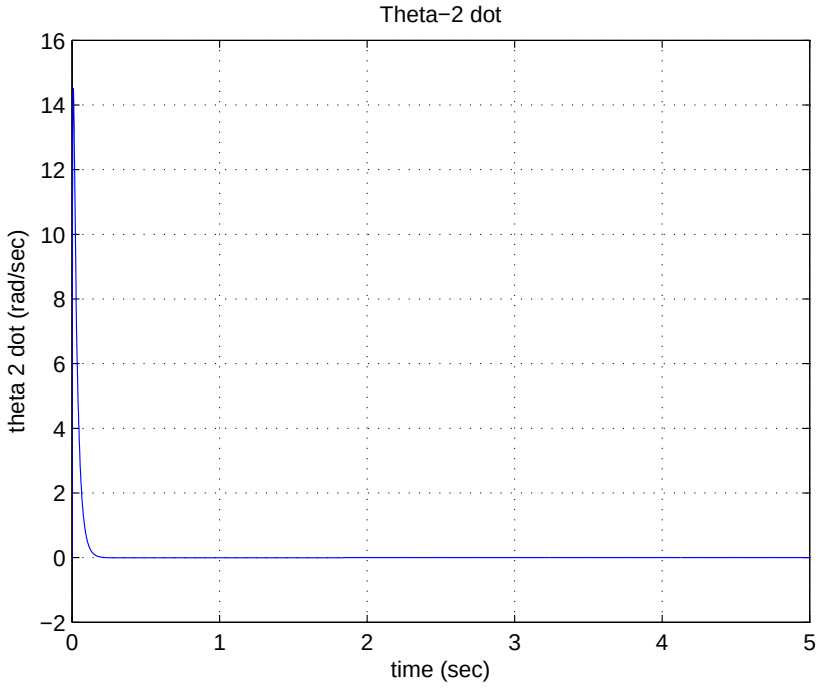


Figure 4.54: Theta 2 dot 'PID control' case 2

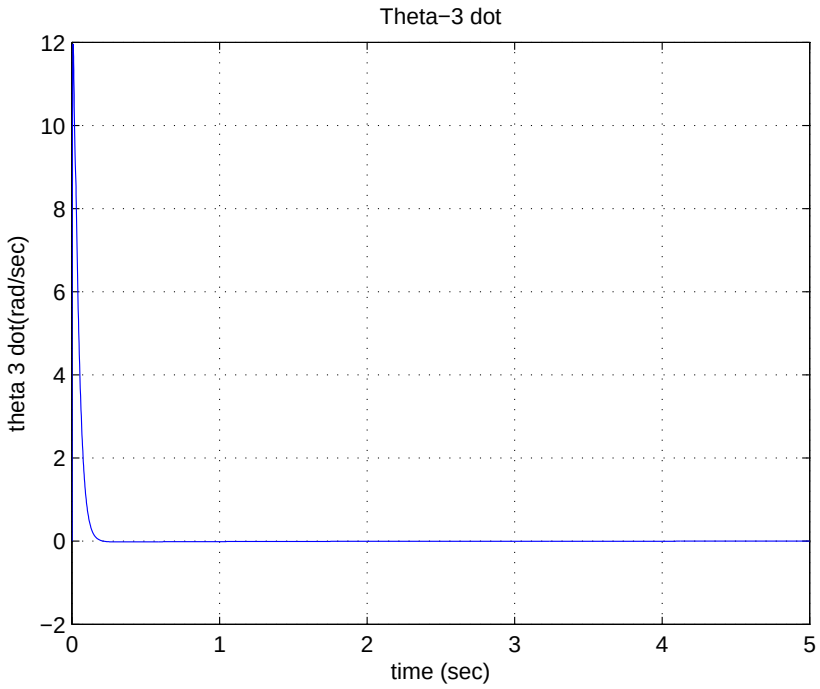


Figure 4.55: Theta 3 dot 'PID control' case 2

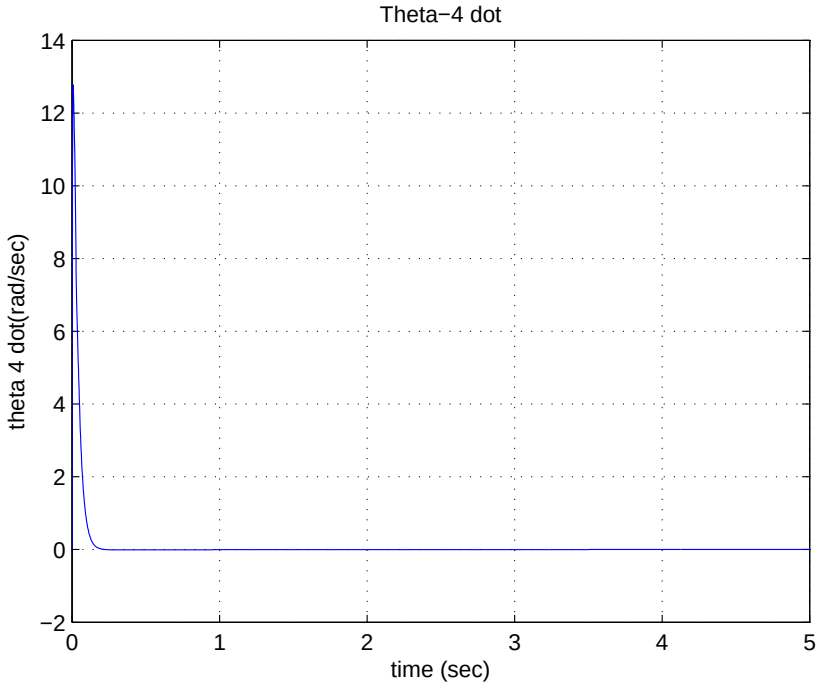


Figure 4.56: Theta 4 dot 'PID control' case 2

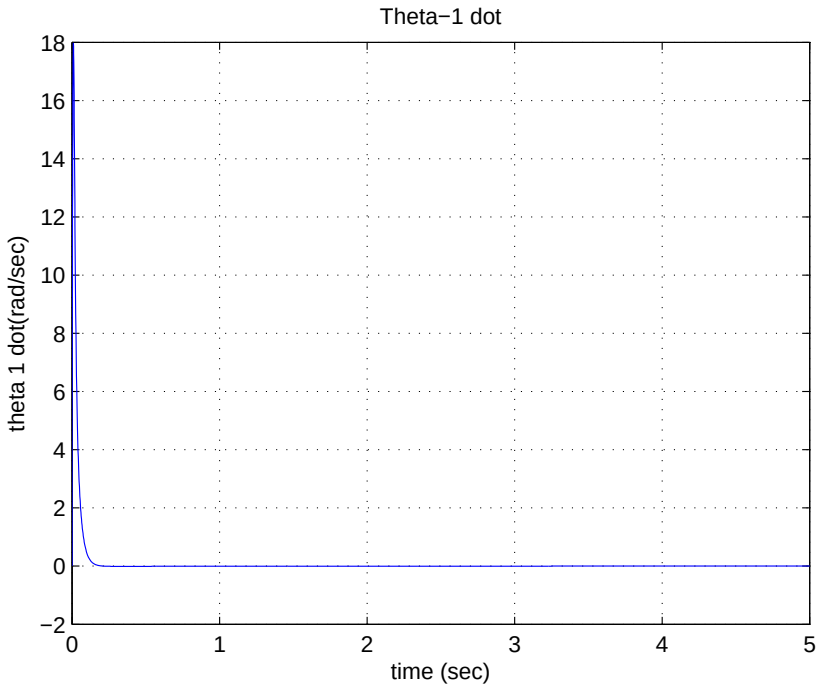


Figure 4.57: Theta 1 dot 'PID control' case 3

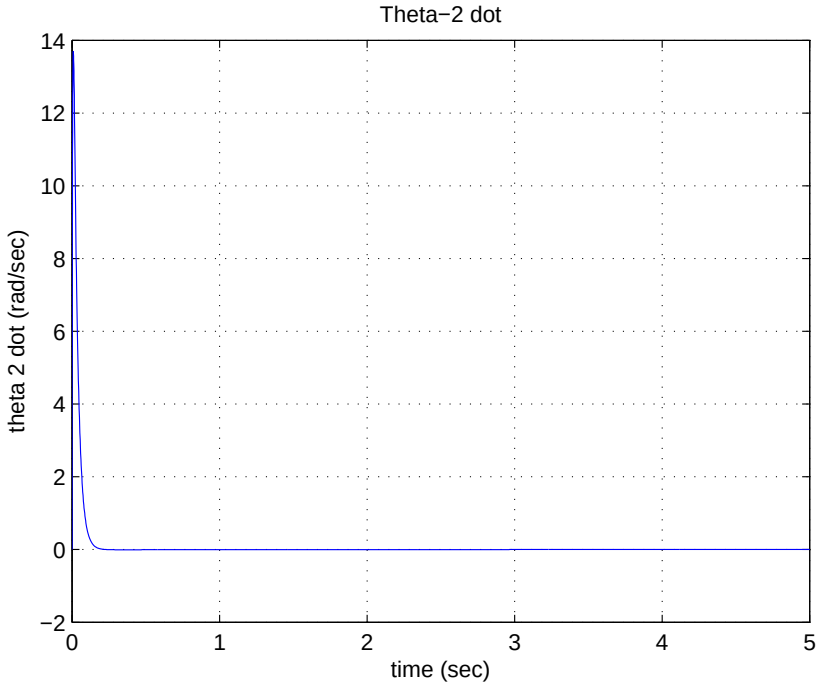


Figure 4.58: Theta 2 dot 'PID control' case 3

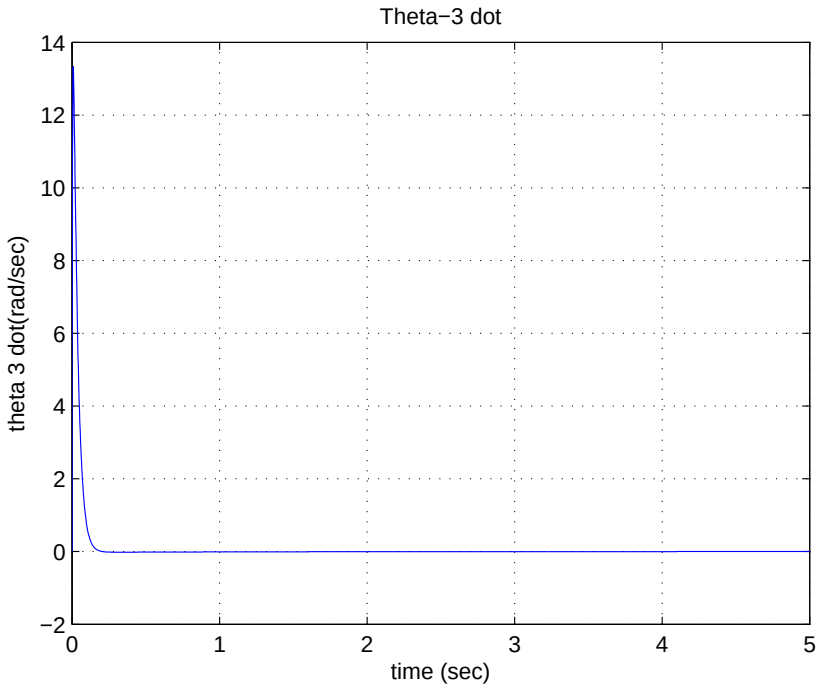


Figure 4.59: Theta 3 dot 'PID control' case 3

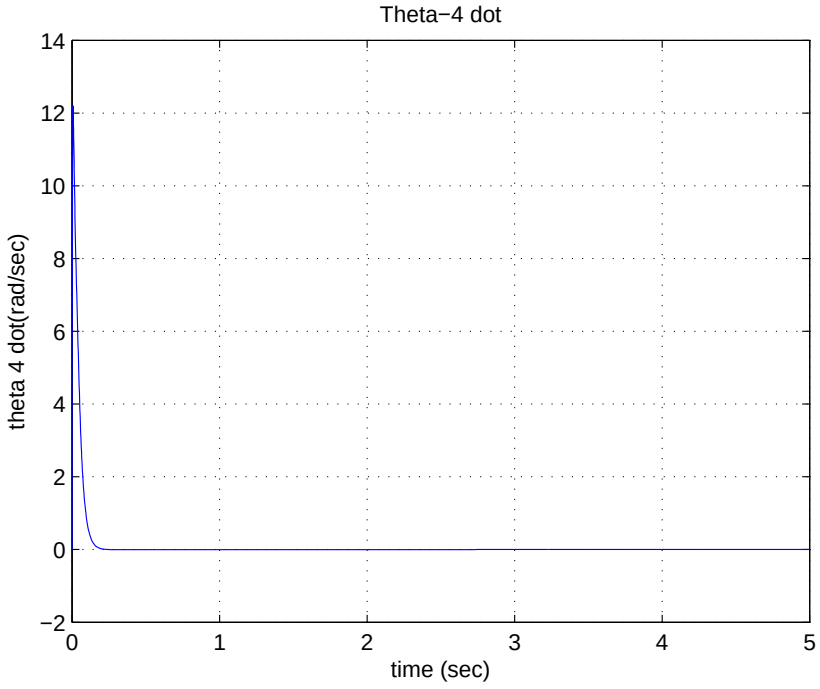


Figure 4.60: Theta 4 dot 'PID control' case 3

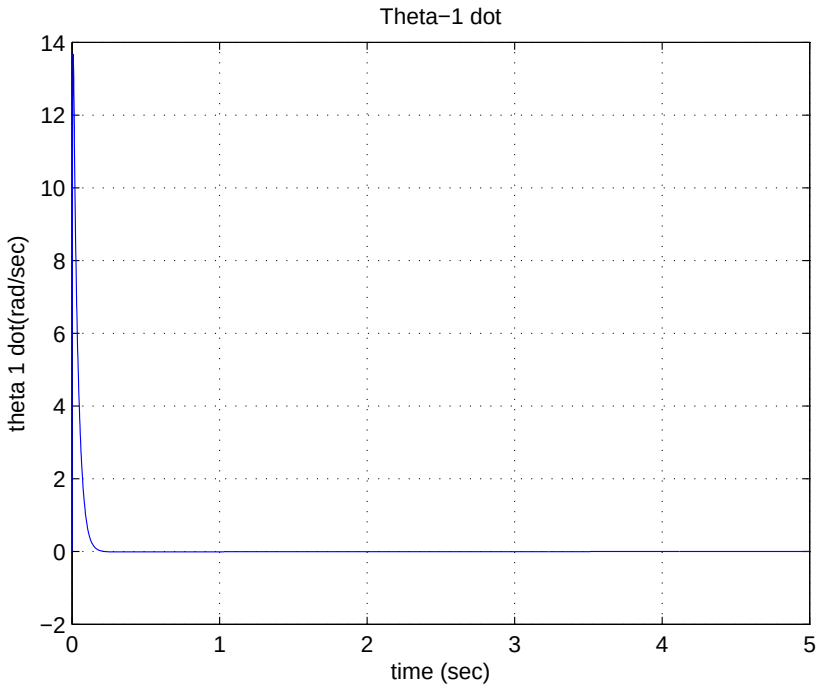


Figure 4.61: Theta 1 dot 'PID control' case 4

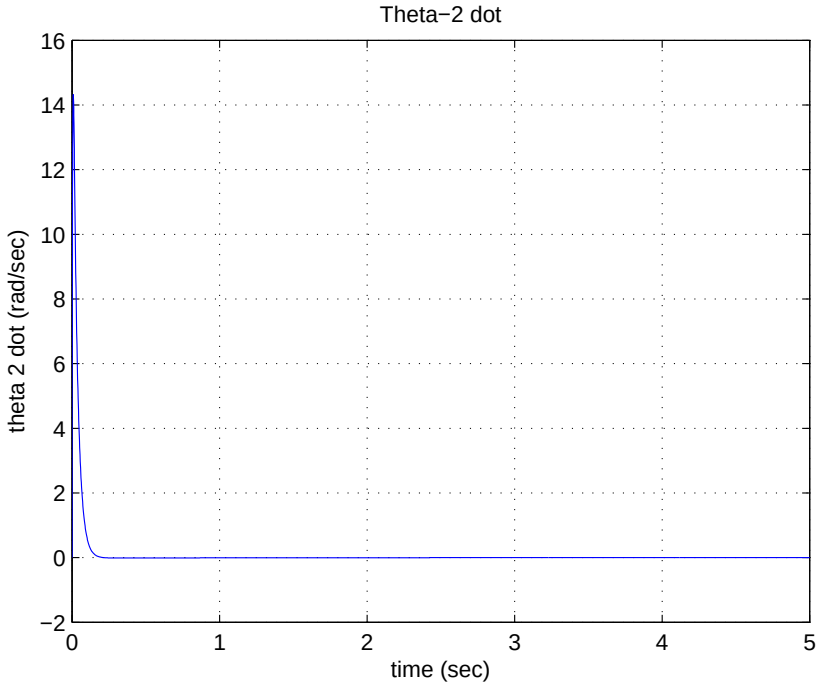


Figure 4.62: Theta 2 dot 'PID control' case 4

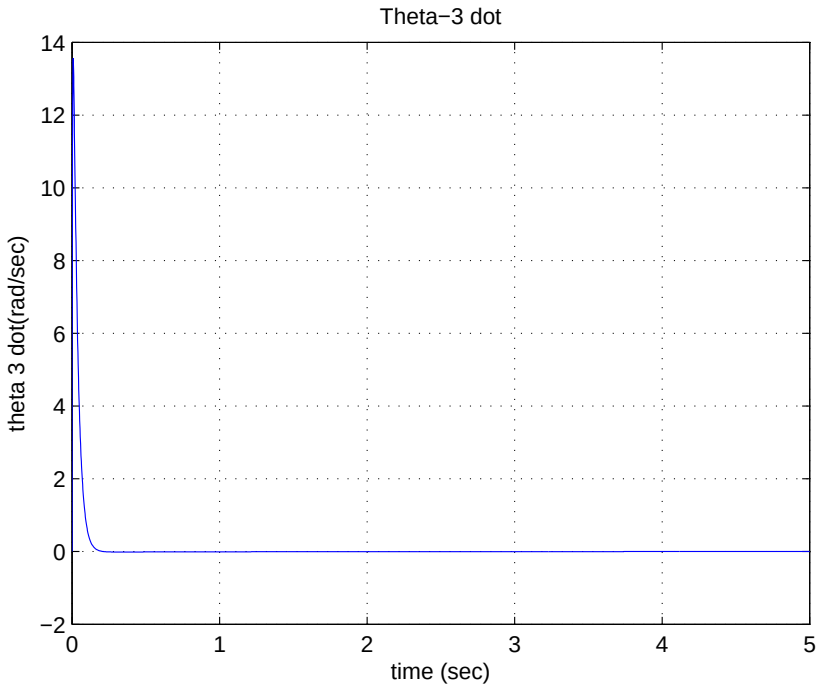


Figure 4.63: Theta 3 dot 'PID control' case 4

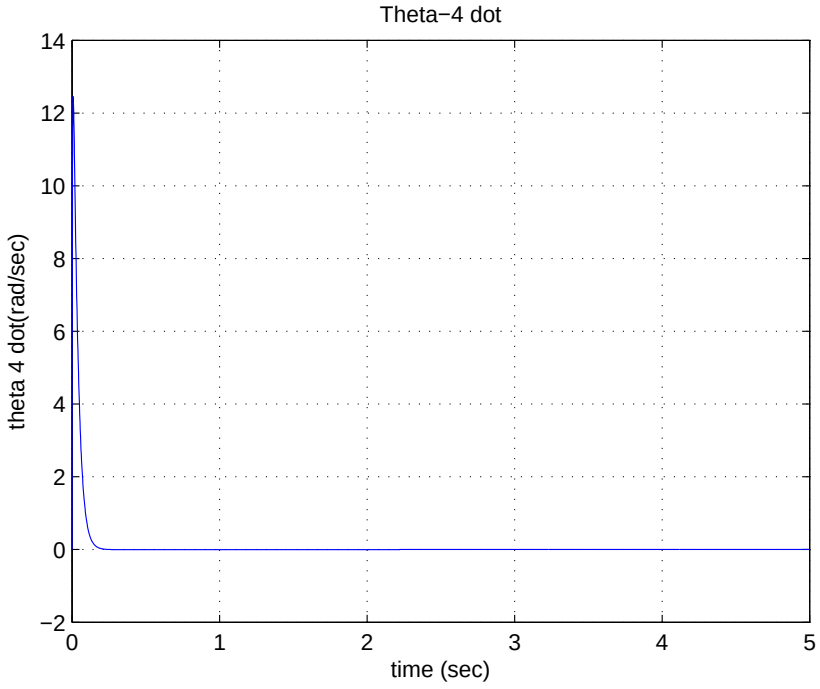


Figure 4.64: Theta 4 dot 'PID control' case 4

The following group of figures depict the time responses of the joints' torques. It is clear that the control inputs (torques) for almost all the joints are very small (approaching zero in around 0.3 sec), which means that all the joints reach the desired position in a very short period of time. However, for the last joint in case 4 there is a small amount of constant torque. This could be interpreted such that there is a high Coriolis effect in this configuration introducing this shift in the control signal.

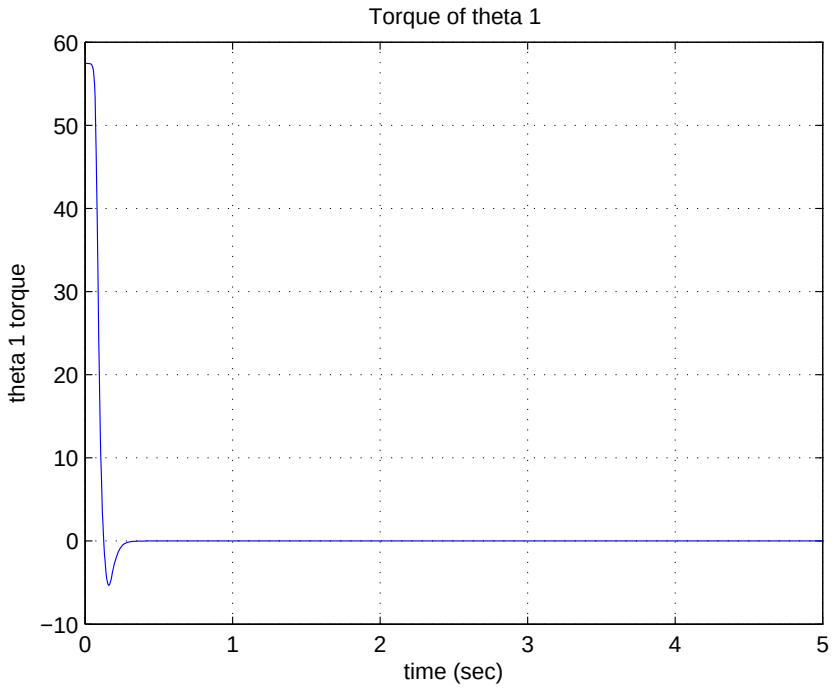


Figure 4.65: Joint 1 Control Signal 'PID control' case 1

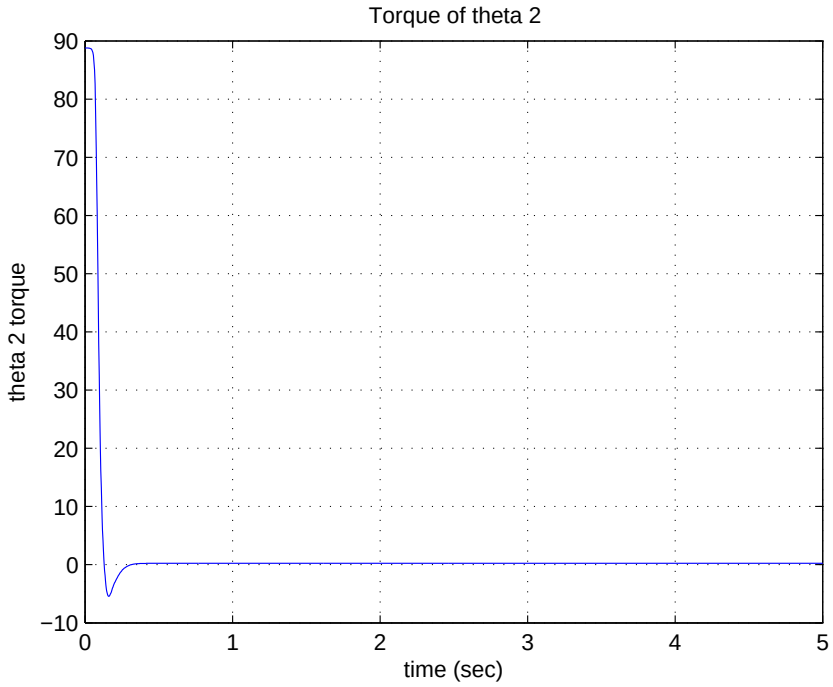


Figure 4.66: Joint 2 Control Signal 'PID control' case 1

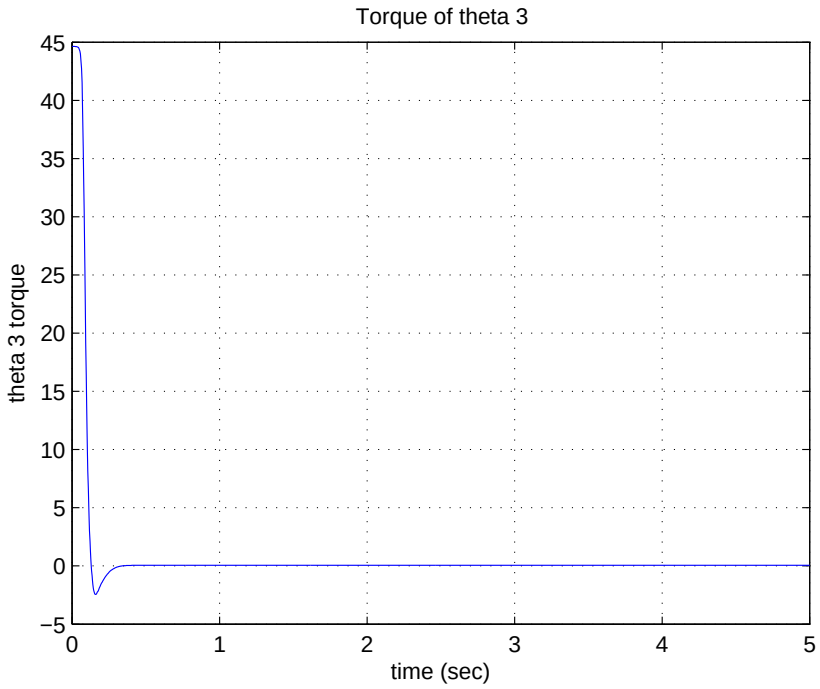


Figure 4.67: Joint 3 Control Signal 'PID control' case 1

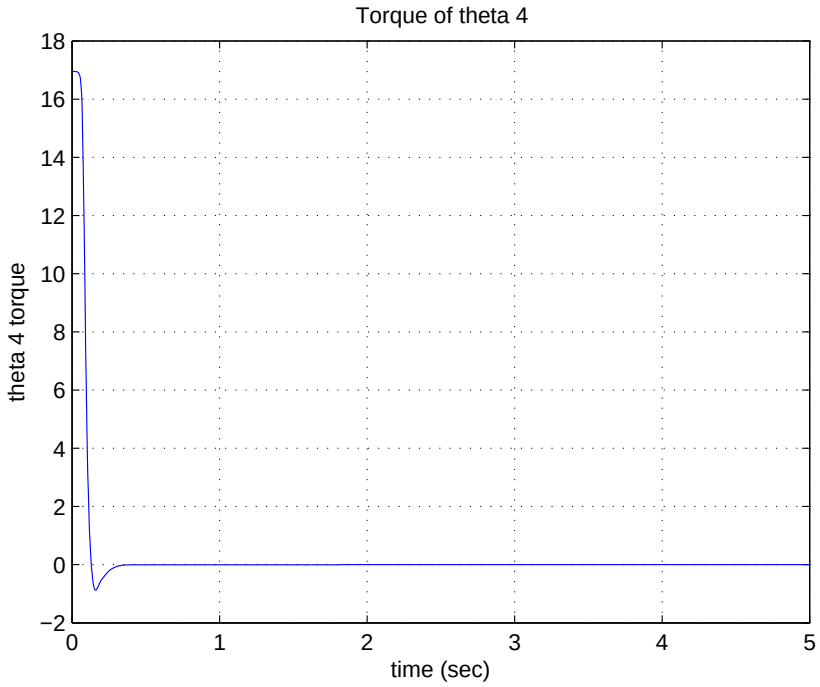


Figure 4.68: Joint 4 Control Signal 'PID control' case 1

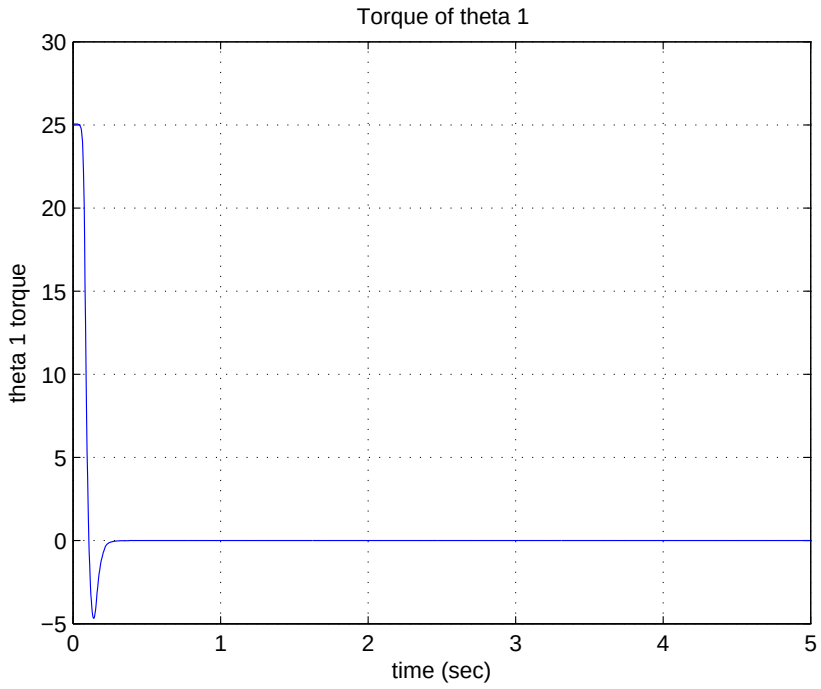


Figure 4.69: Joint 1 Control Signal 'PID control' case 2

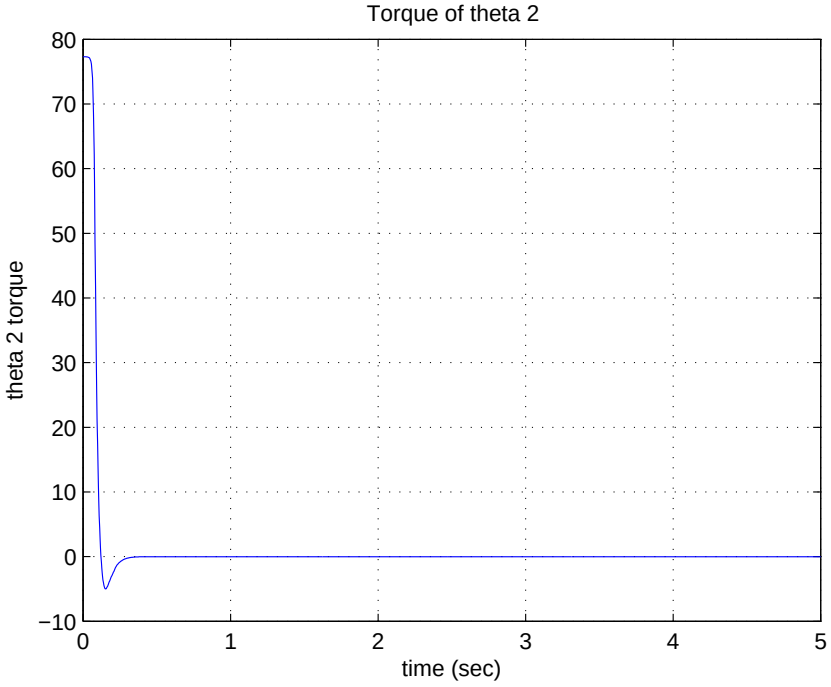


Figure 4.70: Joint 2 Control Signal 'PID control' case 2

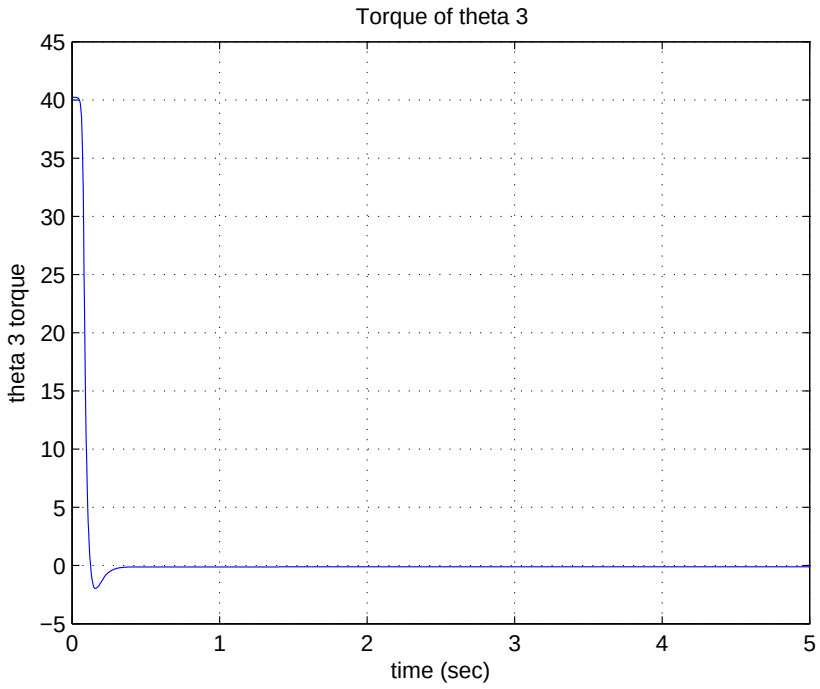


Figure 4.71: Joint 3 Control Signal 'PID control' case 2

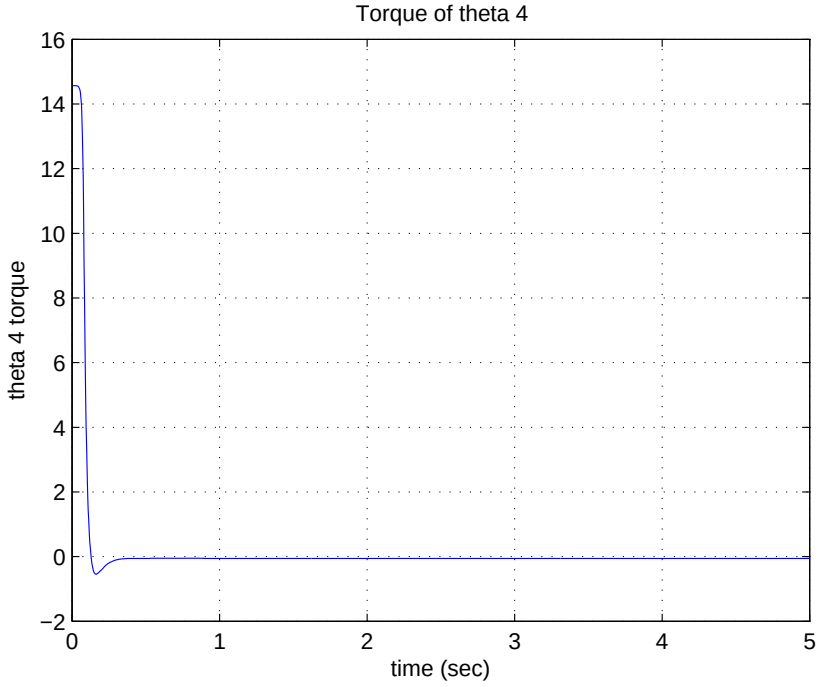


Figure 4.72: Joint 4 Control Signal 'PID control' case 2

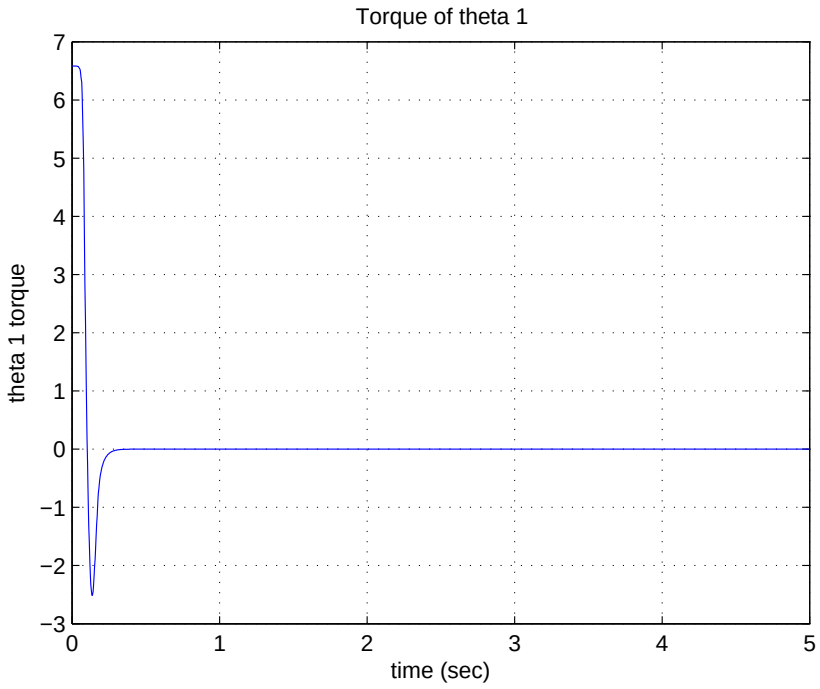


Figure 4.73: Joint 1 Control Signal 'PID control' case 3

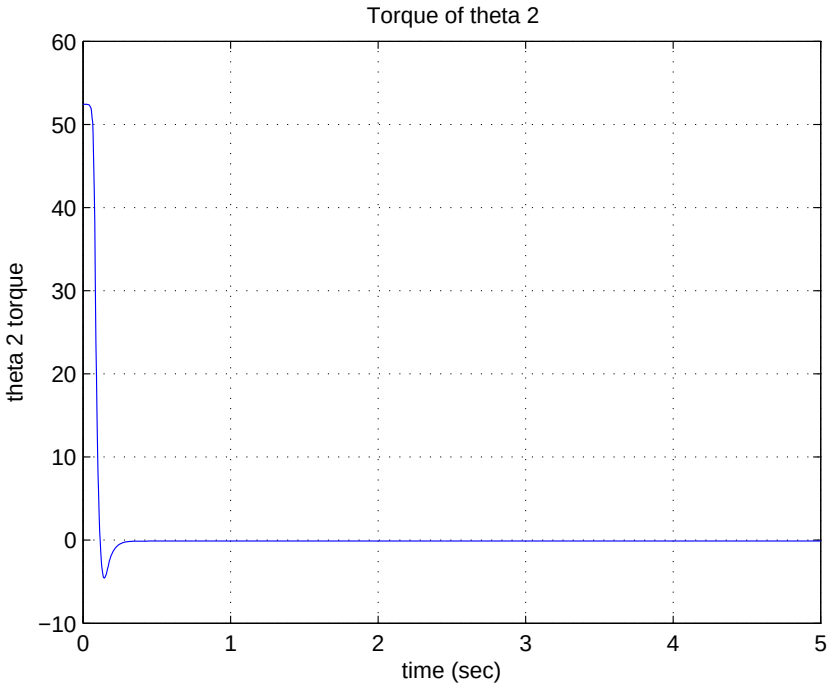


Figure 4.74: Joint 2 Control Signal 'PID control' case 3

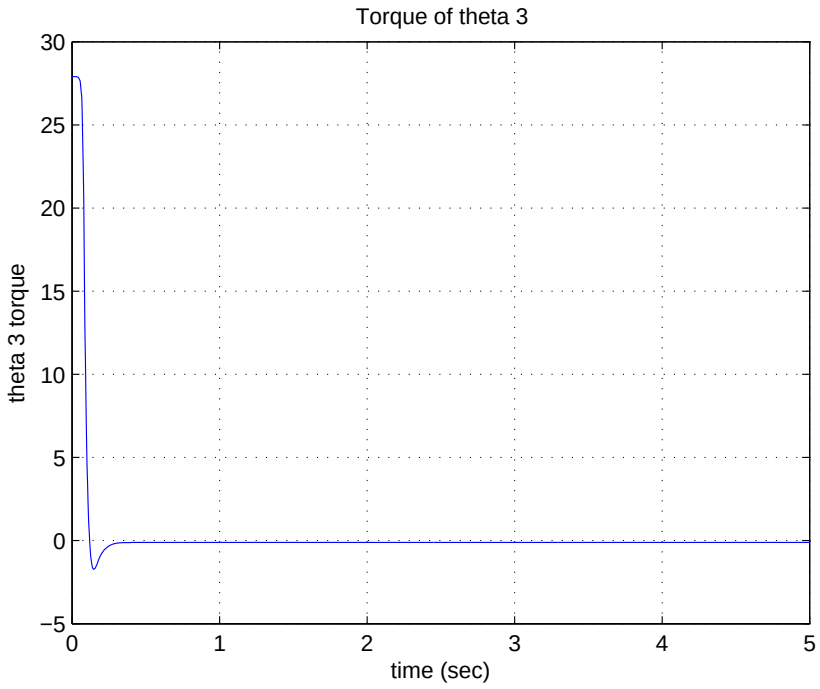


Figure 4.75: Joint 3 Control Signal 'PID control' case 3

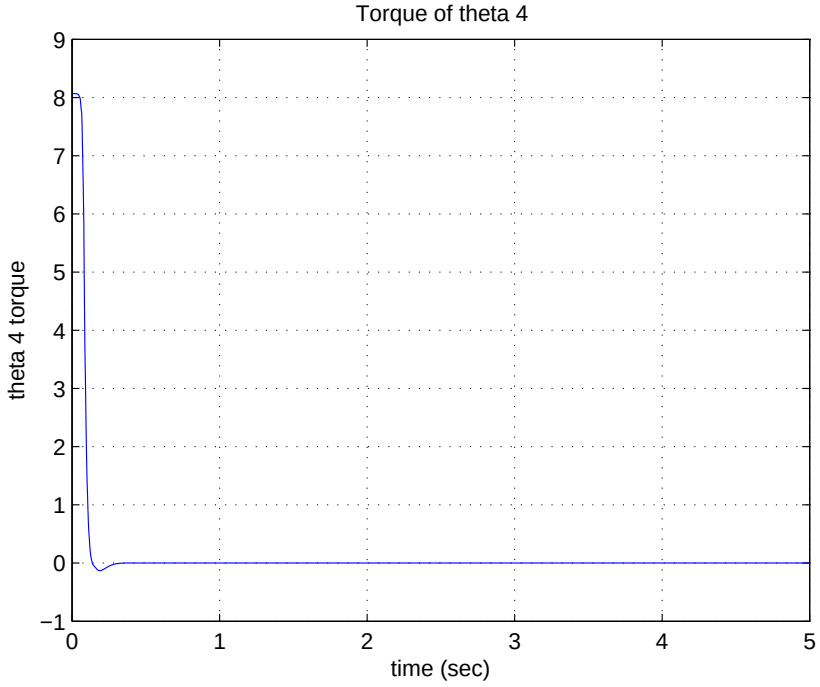


Figure 4.76: Joint 4 Control Signal 'PID control' case 3

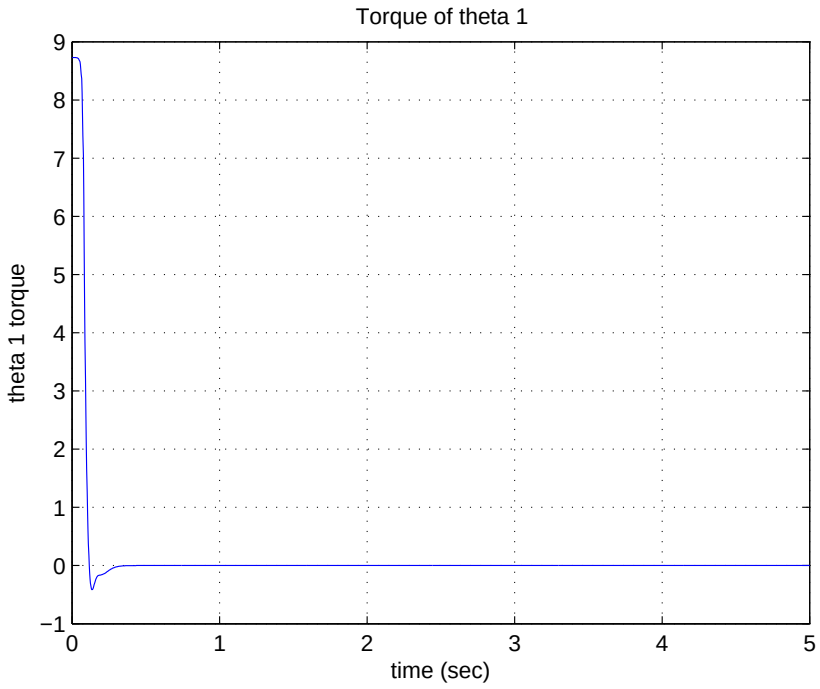


Figure 4.77: Joint 1 Control Signal 'PID control' case 4

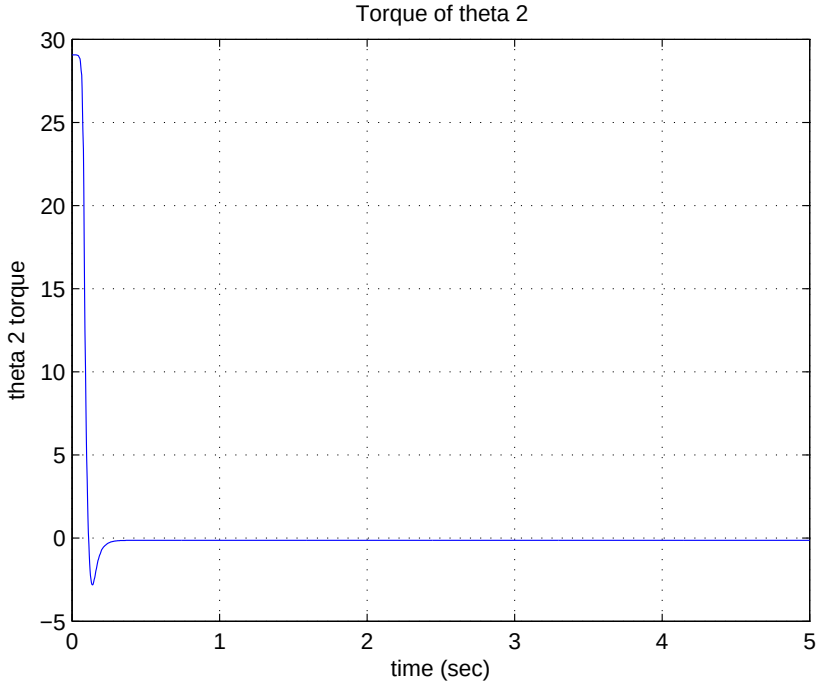


Figure 4.78: Joint 2 Control Signal 'PID control' case 4

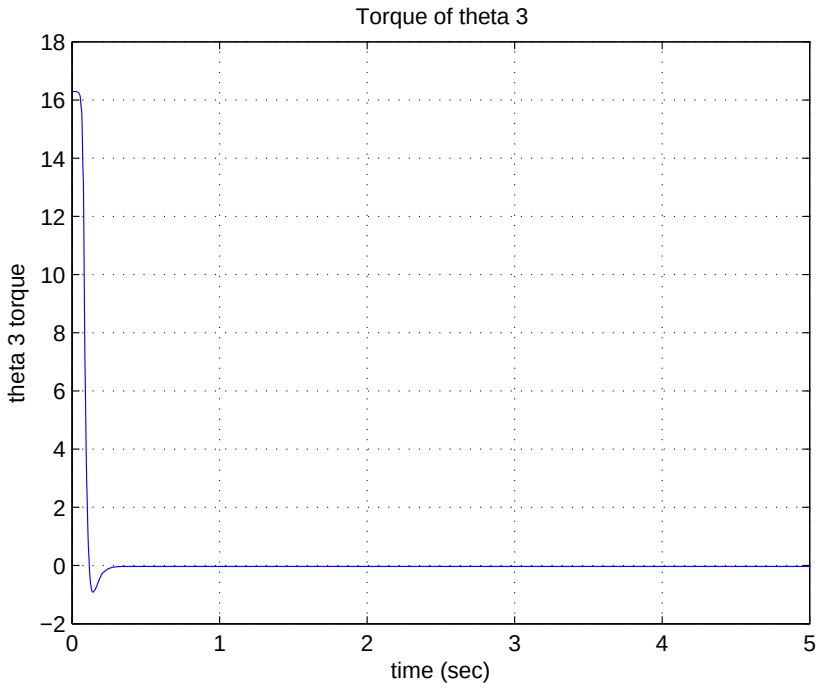


Figure 4.79: Joint 3 Control Signal 'PID control' case 4

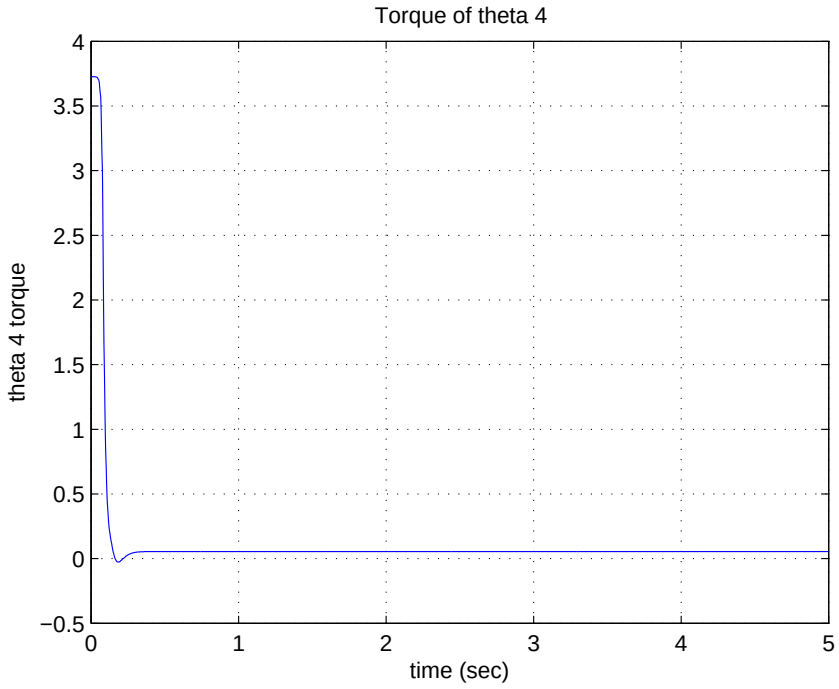


Figure 4.80: Joint 4 Control Signal 'PID control' case 4

4.4 Trajectory Tracking Control

In several practical situations, the joints of the robotic manipulator need to track a time dependent required path to produce a prescribed time dependent route at the gripper. In such circumstances the position control is not satisfactory alone and more sophisticated techniques are required. Instances of such cases include those when the manipulator end-effector must pass through a predetermined track with a specified speed in such situations as 3D objects welding, obstacles avoidance, etc. In these circumstances, the issue is how near the robot can follow a known trajectory, i.e. precise tracking becomes important during the whole journey.

4.4.1 Feedback Linearizion

In this section we introduce the concept of feedback linearizion of nonlinear systems. The basic idea behind the feedback linearizion is to design a nonlinear control law (inner control loop) that, in the classic sense, entirely linearizes the nonlinear system by an appropriate state space coordinate change. One can then design an outer loop control (a second stage) in the current coordinates to fulfill the conventional requirements of control design such as tracking, disturbance rejection and so on [42]. we start our discussion, recalling that dynamic model can be expressed - via Lagrange equation - in the following form

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \tag{4.22}$$

Now, define the new control law u as

$$u = D(\theta)^{-1}(\tau - C(\theta, \dot{\theta})\dot{\theta} - g(\theta) \tag{4.23}$$

Inserting in the general equation of motion 4.22, this new control input produces the

following simple linear structure:

$$\ddot{\theta} = u \tag{4.24}$$

Now, one has to construct a control for this simple system, taking the tracking error as $\tilde{\theta} = \theta^d - \theta$. We can show that the control law:

$$u = \ddot{\theta}^d - 2\lambda\dot{\tilde{\theta}} - \lambda^2\tilde{\theta} \tag{4.25}$$

with $\lambda \geq 0$, leads to an exponentially stable closed-loop dynamics. By substituting (4.25) into (4.24) the closed-loop error dynamics can be derived as:

$$\ddot{\theta} + 2\lambda\dot{\tilde{\theta}} + \lambda^2\tilde{\theta} = 0 \tag{4.26}$$

After the control law is designed for the new input of the linear-looking system (4.24), the designed new control law can be converted to the initial control using equation (4.22)

$$\tau = D(\theta)u + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \tag{4.27}$$

Formula 4.27 is noted as the "computed torque" in the literature of robotics . This considers naturally that the model utilized in (4.27) is a precise dynamic model. In case there are uncertainties in obtaining the robot dynamical parameters, or the robot dynamic specifications vary, the controller performance is affected adversely [42] . It is important to realize the fact that the computed torque depends on the inversion of the robot dynamics, and that is why it is also known as the inverse dynamics control [27]. Once the dynamic model is at hand, the feedback control law can be obtained by simply substituting equation (4.25) into (4.27):

$$\tau = D(\theta)(\ddot{\theta}^d - 2\lambda\dot{\tilde{\theta}} - \lambda^2\tilde{\theta}) + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \tag{4.28}$$

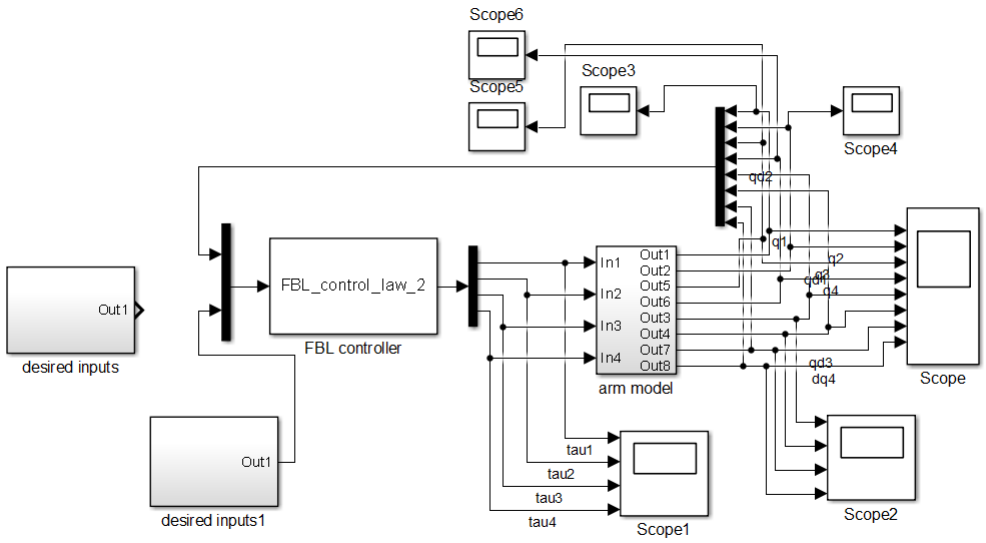


Figure 4.81: Simulink Model of FBL control

Using the MATLAB SIMULINK, we can build a model for the feedback linearization control utilizing some Simulink Blocks.

Both position control and trajectory tracking control are investigated here.

FBL Position Control

For the position control using the FBL, figures (4.82- 4.85) depict the joint angles for position control, the aim here is to move all the joints from the rest position, to 30 degree configuration (as an example). It is obvious that the controller is able to accurately locate the joints at the desired position without any steady-state errors in this case, which shows a better performance compared to that of the PD position controller presented before. The controller gain λ is taken as 100 in this case. Moreover, figures (4.86-4.89) show the joint speeds, and the joint torques are depicted in (4.90-4.93).

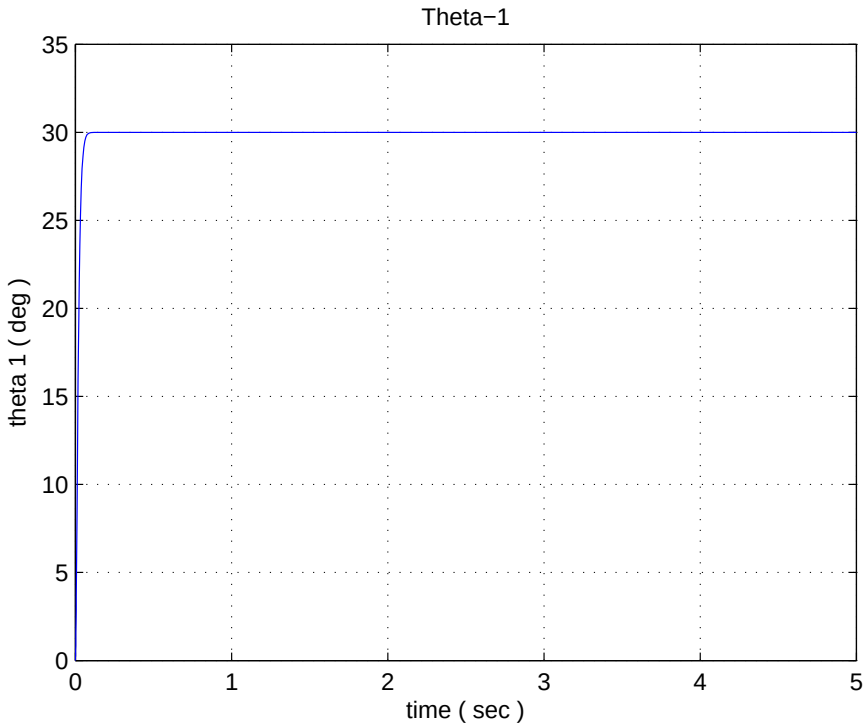


Figure 4.82: Theta 1 Response 'FBL control'

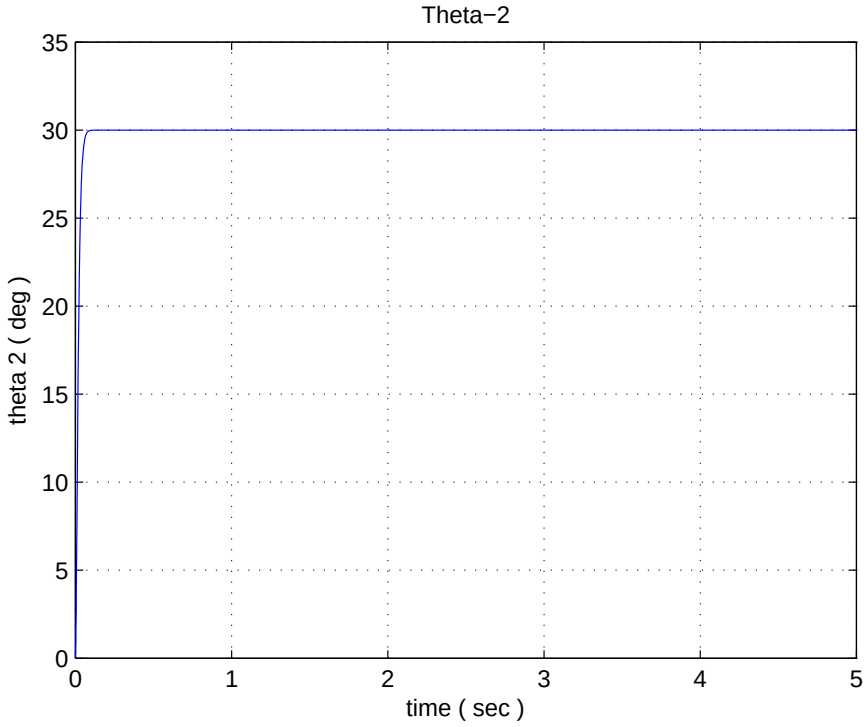


Figure 4.83: Theta 2 Response 'FBL control'

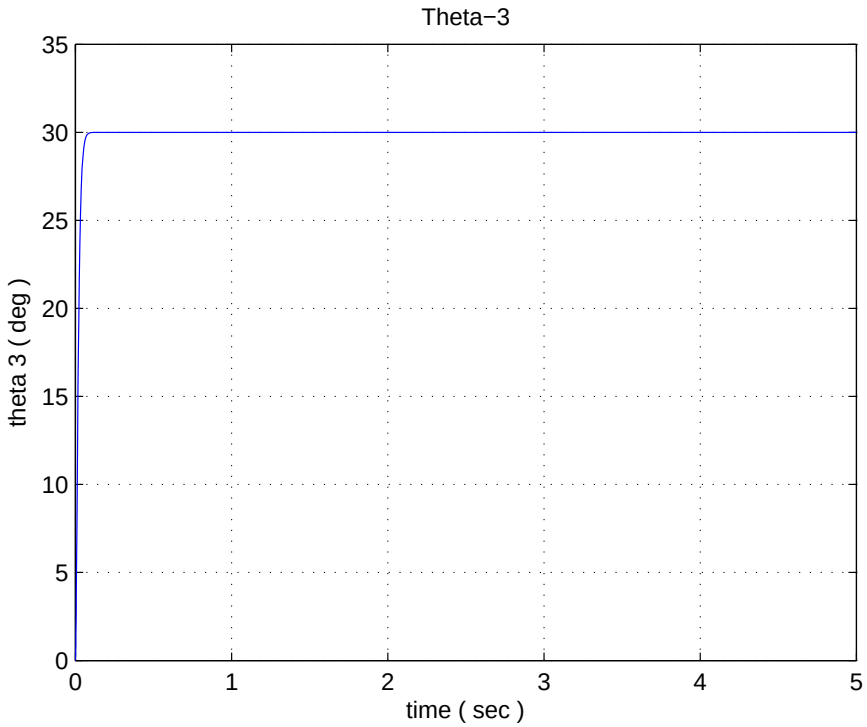


Figure 4.84: Theta 3 Response 'FBL control'

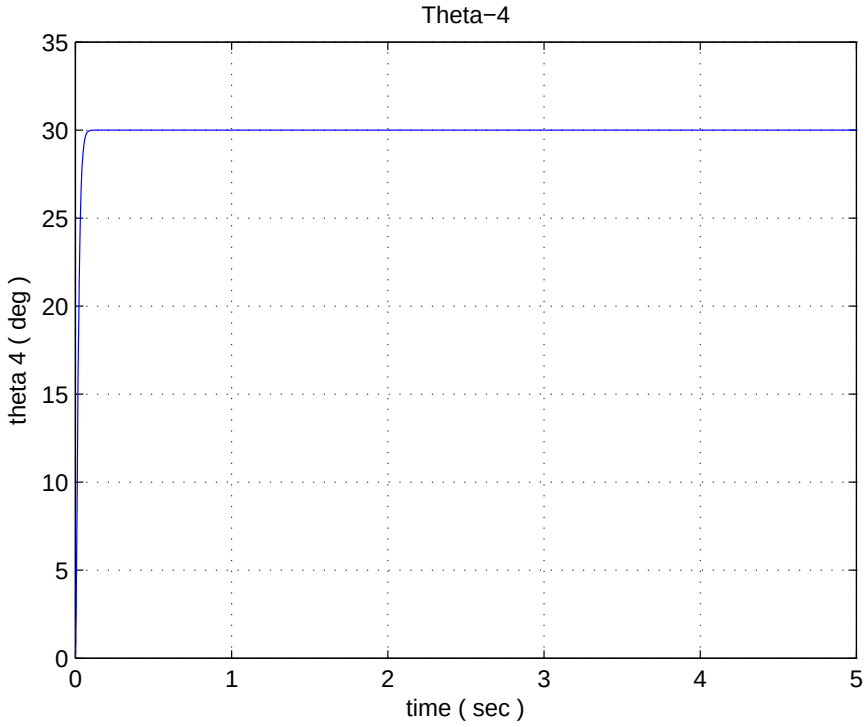


Figure 4.85: Theta 4 Response 'FBL control'

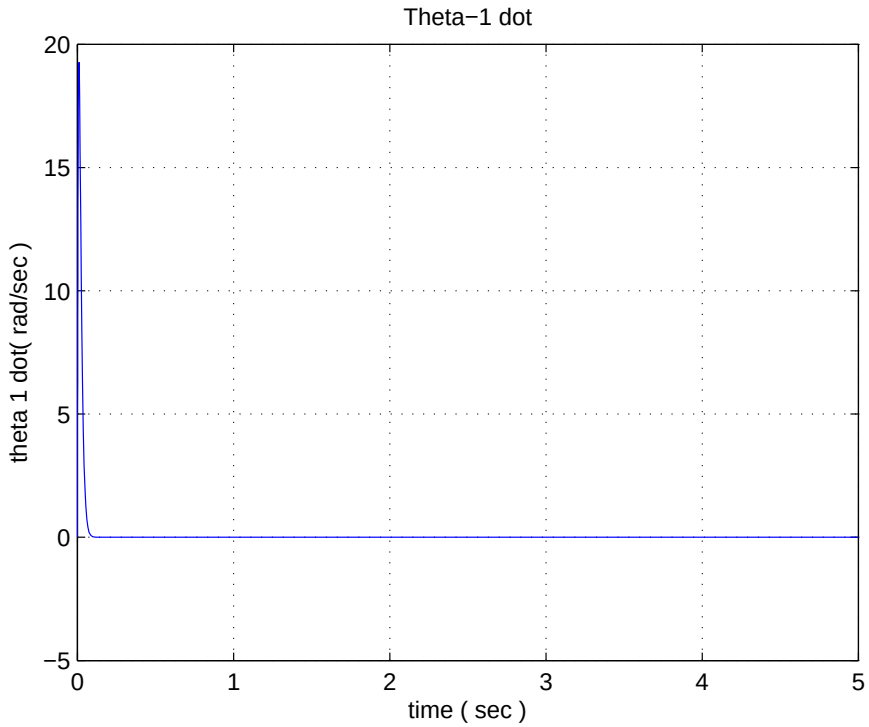


Figure 4.86: Theta dot 1 Response 'FBL control'

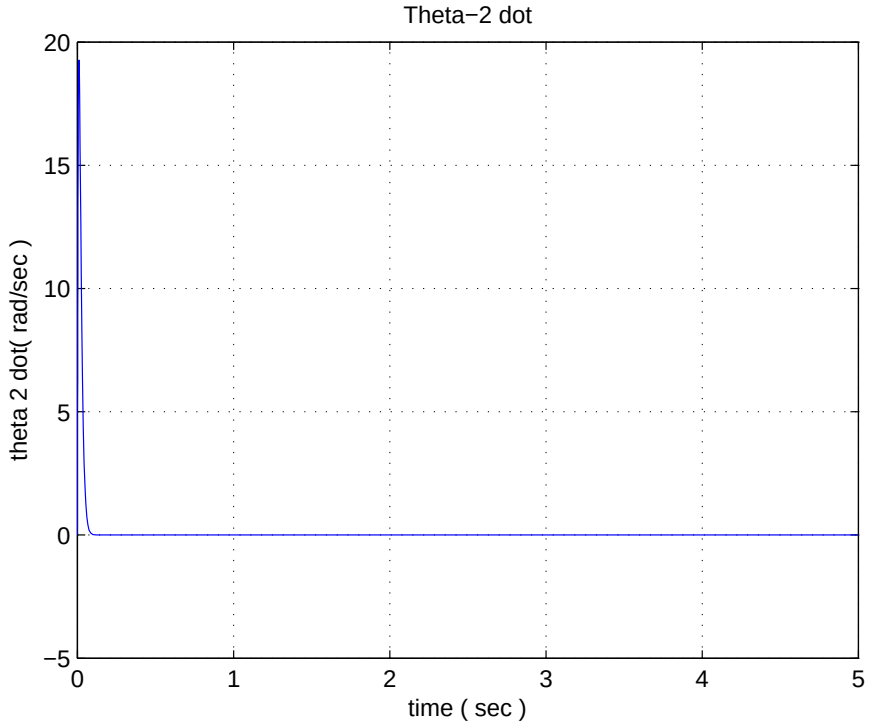


Figure 4.87: Theta dot 2 Response 'FBL control'

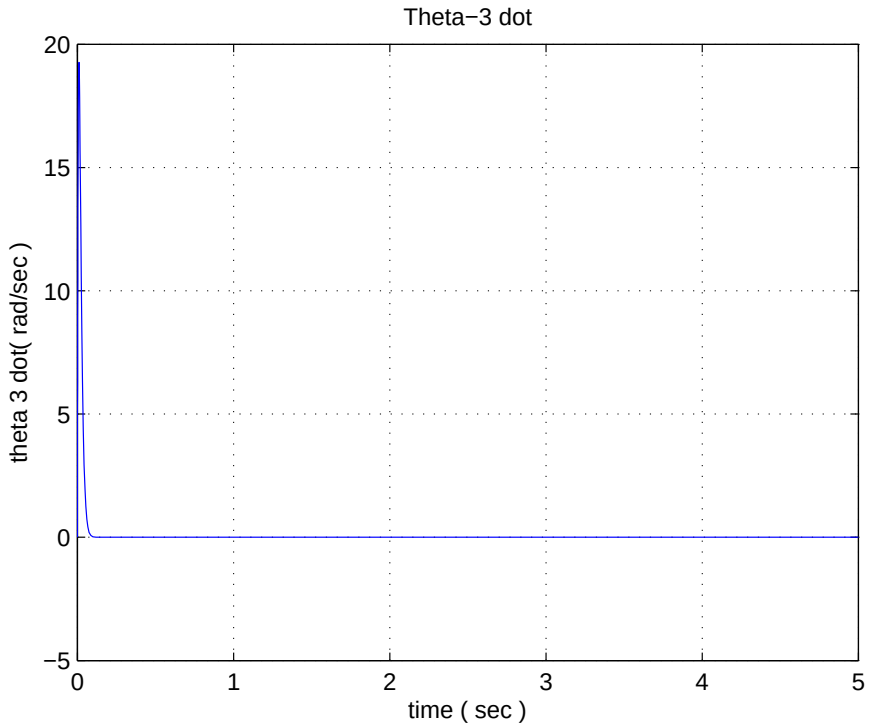


Figure 4.88: Theta dot 3 Response 'FBL control'

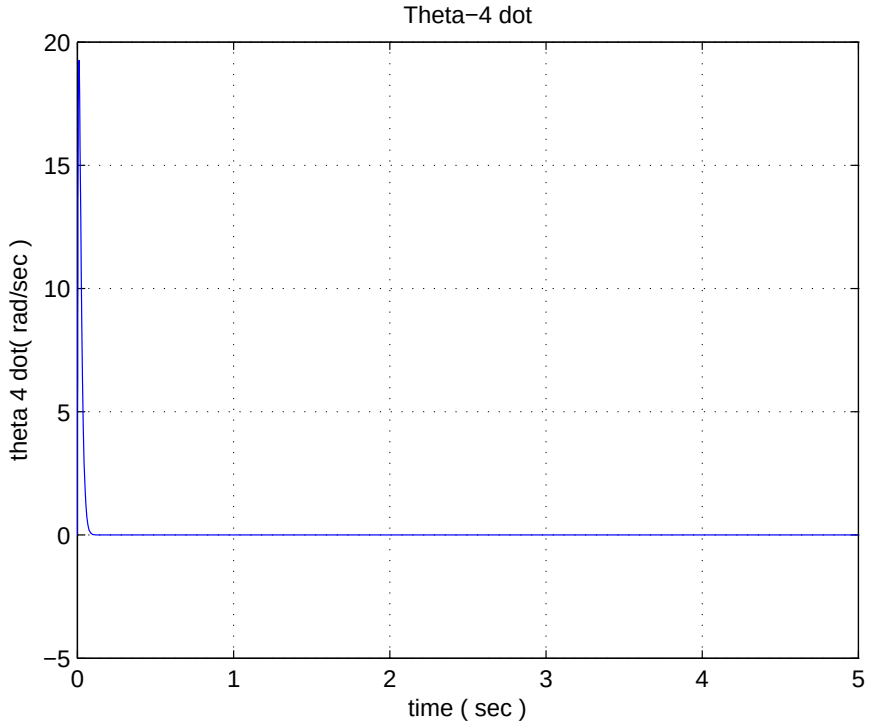


Figure 4.89: Theta dot 4 Response 'FBL control'

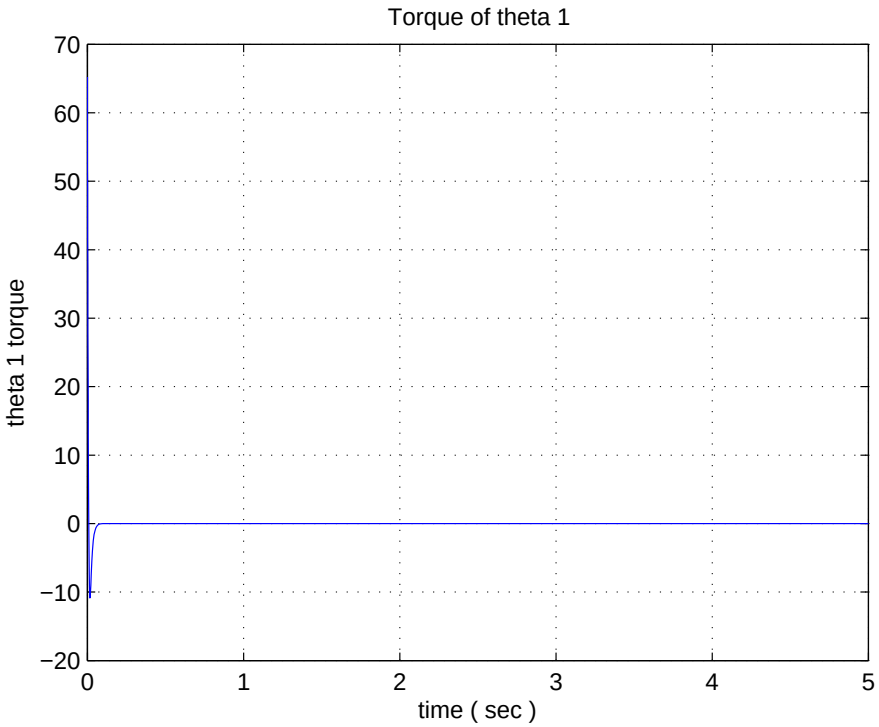


Figure 4.90: Theta 1 torque 'FBL control'

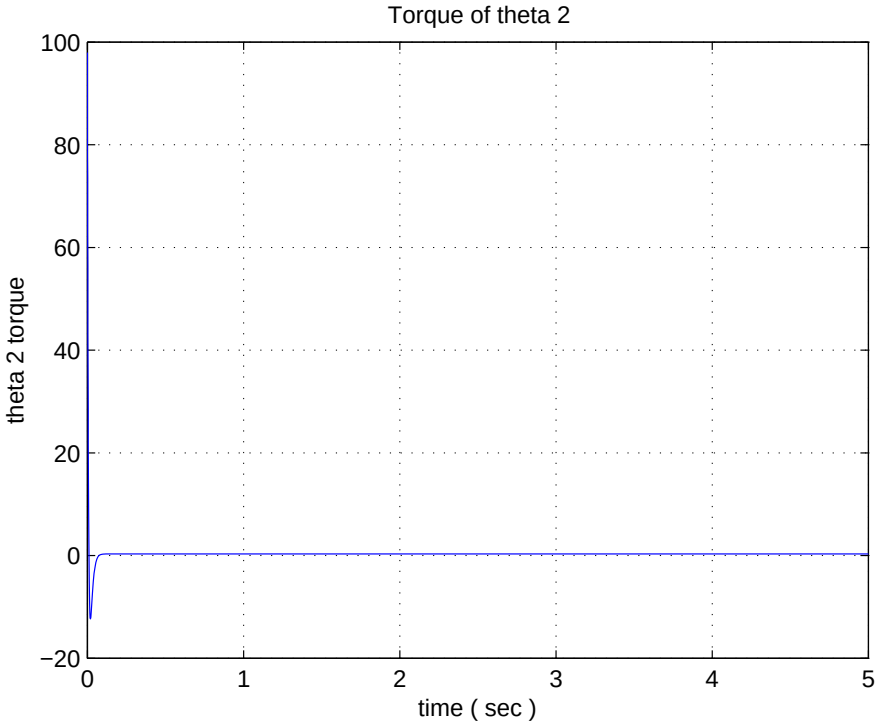


Figure 4.91: Theta 2 torque 'FBL control'

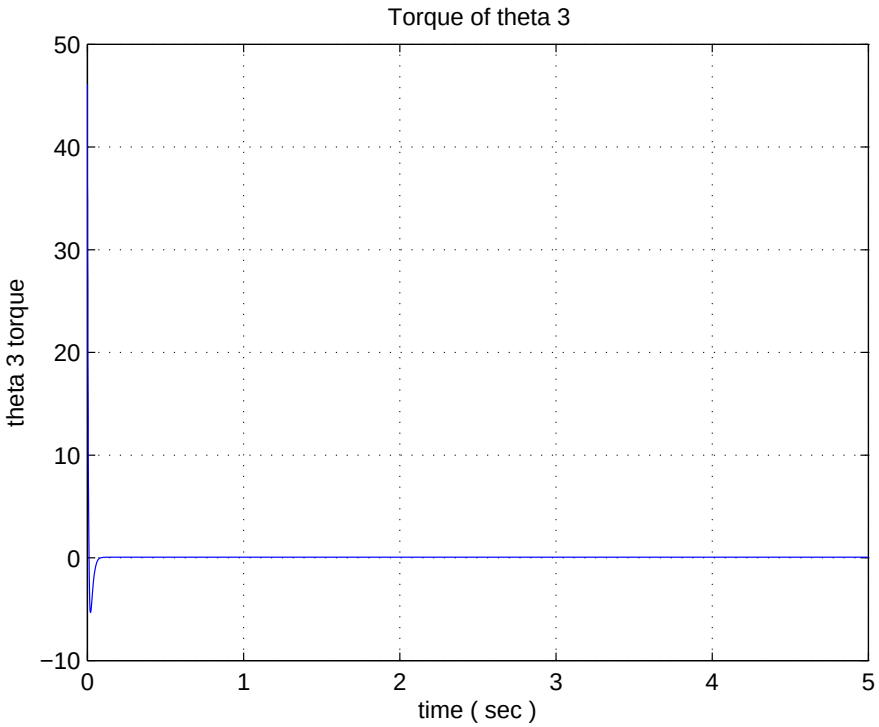


Figure 4.92: Theta 3 torque 'FBL control'

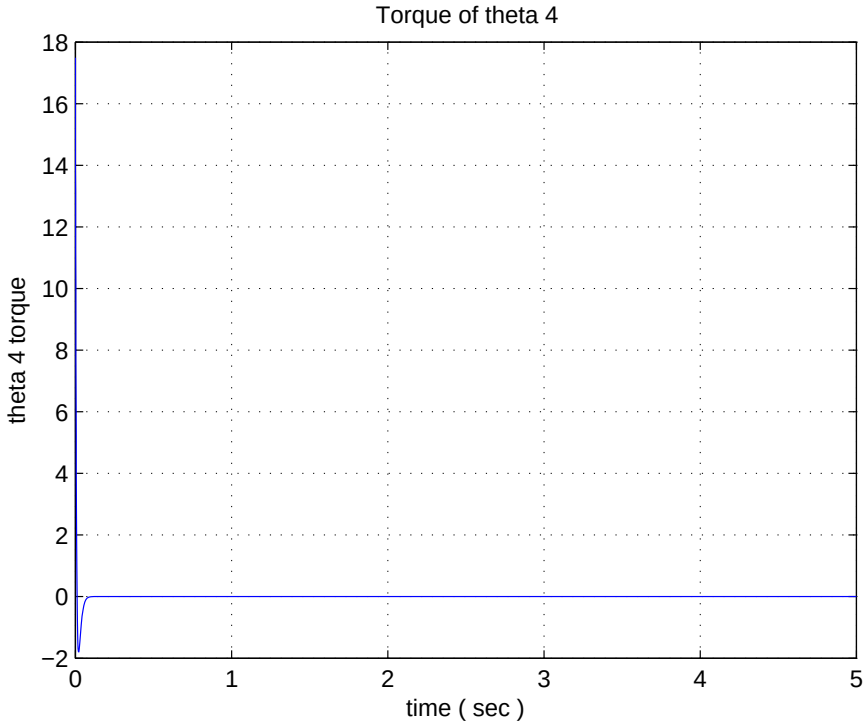


Figure 4.93: Theta 4 torque 'FBL control'

Comparison Between PID Control & Position Control of FBL

Comparing the results of position control using FBL figures (4.82- 4.85) with those (figures 4.27-4.30) for the first case of the PID control (section 4.3.6), as table (4.3) presents, it is clear that for all the angles there is no overshoot in the case of the FBL technique and the settling is 0.0585 sec which is much less than that of the PID control. The rise time for all angles in both approaches is less 0.1 sec.

Table 4.3: Results Comparison

Theta	Controller	Overshoot	Settling time	Rise time
θ_1	PID	2.2940	0.4007	0.0563
	FBL	0.0004	0.0585	0.0336
θ_2	PID	1.5022	0.0976	0.0583
	FBL	0.0004	0.0585	0.0336
θ_3	PID	3.1471	0.7505	0.0638
	FBL	0.0004	0.0585	0.0336
θ_4	PID	3.3943	0.8327	0.0606
	FBL	0.0004	0.0585	0.0336

We can conclude that results obtained with the FBL control are better than those of the PID control.

FBL Trajectory Tracking Control

The following figures show the controlled paths using the feedback linearization technique. It is apparent from figures (4.94-4.97) that the FBL control is capable of drawing the joint variables in the prescribed path, which agrees with the corresponding scenario for the position control, using the same gain ($\lambda = 100$) as before.

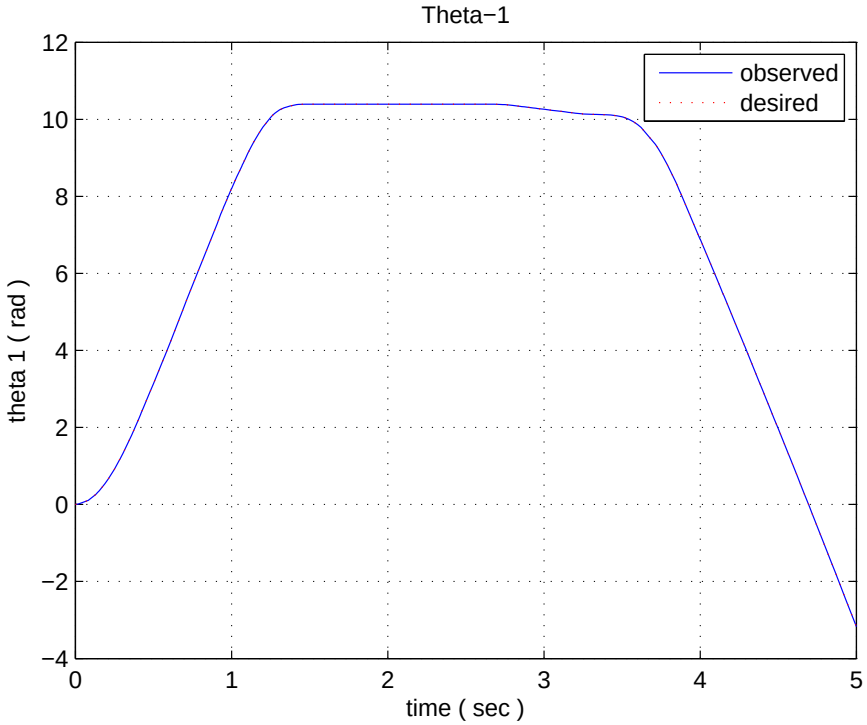


Figure 4.94: Theta 1 Response trajectory tracking

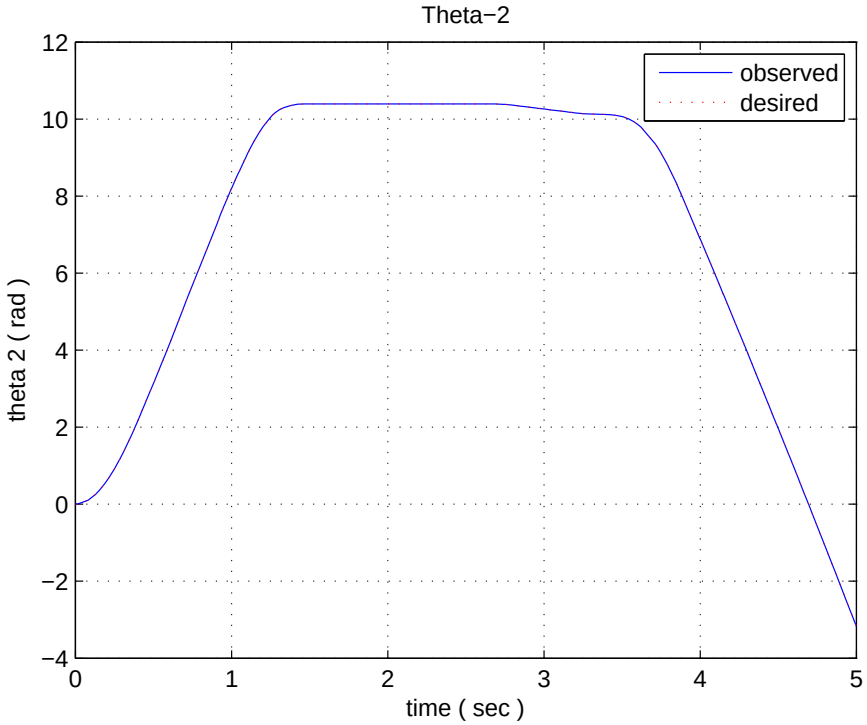


Figure 4.95: Theta 2 Response trajectory tracking

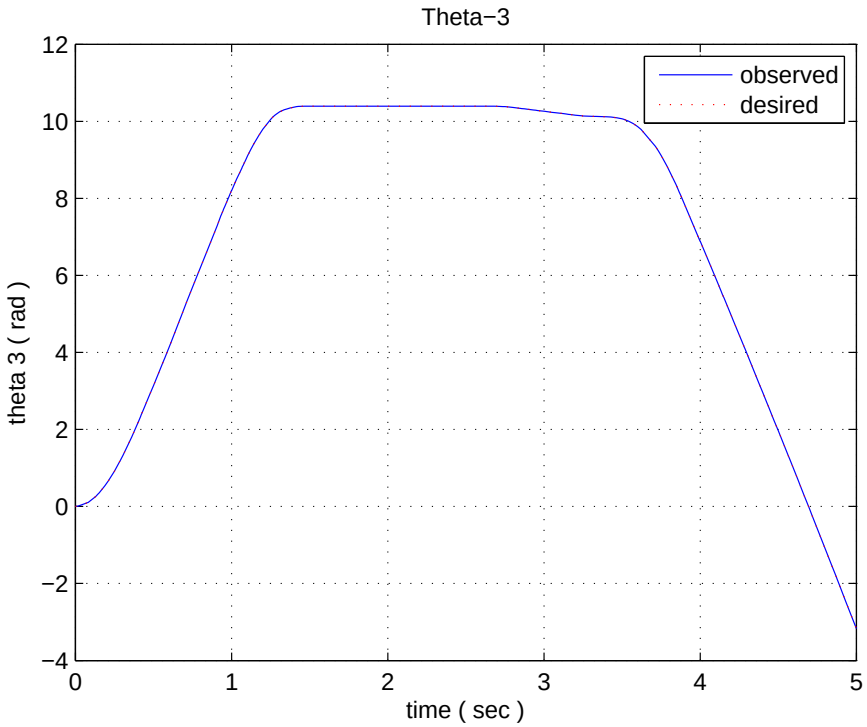


Figure 4.96: Theta 3 Response trajectory tracking

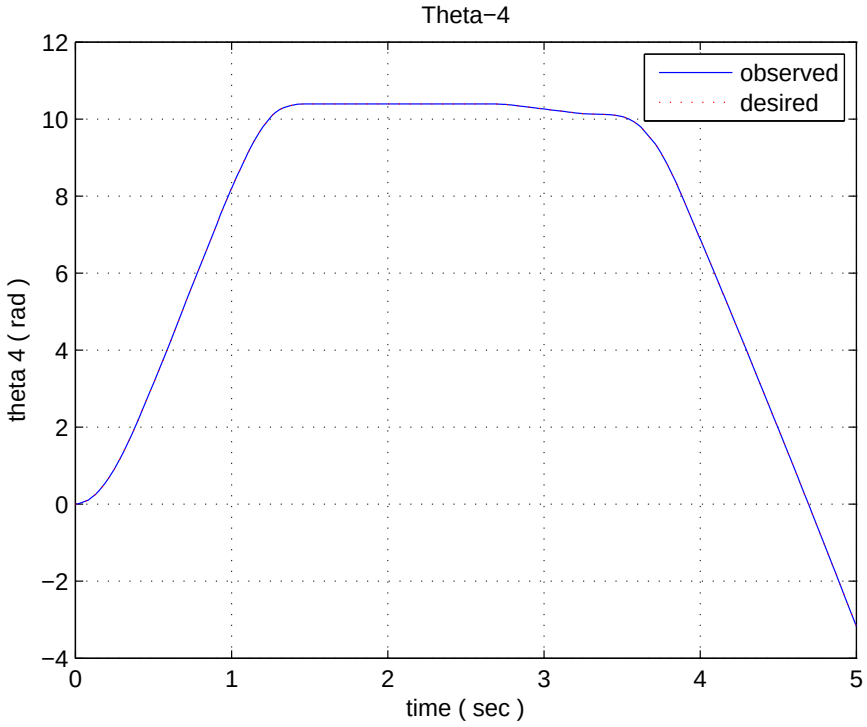


Figure 4.97: Theta 4 Response trajectory tracking

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

This chapter includes some conclusion as a result of this work and highlights some recommendations for future extension.

5.1 Conclusions

Following remarks can be made as a result of this work:

- Both the DH convention and the PE formulation are implemented in the specific robotic manipulator in this work. Based on the implementation, both the formulations are found to be resulting in the same kinematics equations.
- Angles computed using the inverse kinematics are tested on the robotic manipulator using the Java language and are physically realized.
- It is found out that for the PD control, the manipulator's joint velocities disappear at the steady state, indicating that the manipulator remains at a state of equilibrium at the final state.
- Using the DE algorithm, the PID parameters are tuned to obtain a satisfactory manipulator response in terms of the rise and settling times, overshoot and

steady state error.

- For the PID control, the gravity force has a great effect on the overshoot and settling time of various links. It is observed that when the gravity force is high, then the overshoot of the corresponding link is expected to be high.
- The further from the origin the link is, the higher the overshoot and settling time are expected.
- The rise time, on the other hand, is proportional to the centrifugal force. In other words, when the centrifugal force increases, the rise time also increases. However, when the link reaches its steady-state position, then the corresponding rise time decreases.
- Although both the controllers (PID & FBL) are used effectively in controlling the arm, the FBL controller is recognized to have superior robust performance over the PID controller.

5.2 Recommendations

Based on the above conclusion we recommend that:

- Physical testing of the manipulator control by real time experiments would prove to be useful.
- Various other control methods can be studied for a more comprehensive coverage.
- Computer simulation and animation of the controlled behavior would be beneficial for visualization.
- Including the dynamics of the servo in the dynamic model would open other directions of research.

REFERENCES

[1] R. N. Jazar, *Theory of Applied Robotics*. Boston, MA: Springer US, 2010. [Online]. Available: http://link.springer.com/10.1007/978-1-4419-1750-8

[2] M. Matarić, *The Robotics Primer*, ser. Intelligent robotics and autonomous agents. MIT Press, 2007. [Online]. Available: https://books.google.com.sa/books?id=WWJPjgz-jgEC

[3] "robot." [Online]. Available: http://www.zmescience.com/tag/robot/

[4] K. E. Clothier and Y. Shang, "A geometric approach for robotic arm kinematics with hardware design, electrical design, and implementation," *Journal of Robotics*, 2010. [Online]. Available: http://search.proquest.com/docview/856031070/abstract?accountid=27795#

[5] Barinka, Lukas, and Roman Berka, "Barinka, lukas, and roman berka. "inverse kinematics-basic methods." web.< http://www. cescg. org/CESCG-2002/LBarinka/paper. pdf (2002)." 2002.

[6] "Aristidou, andreas, and joan lasenby. "inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver." (2009)."

[7] D. V. K. Banga, Jasjit Kaur, "Jasjit kaur, dr. v k banga "simulation of robotic arm having three link manipulator", international journal of research in engineering and technology (IJRET) vol. 1 no. 2 march, 2012 ISSN: 2277-4378, 2012." 2012.

[8] "Zulfiqar ali soomro, " kinematic modeling and simulation of 2-d link and 3-r pendulum serial manipulator robotics arm" international researcher volume no.2 issue no4. december 2013." 2013.

[9] D. Kostic, B. de Jager, M. Steinbuch, and R. Hensen, "Modeling and identification for high-performance robot control: an RRR-robotic arm case study," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 6, pp. 904–919, Nov. 2004.

[10] D. M. A. Ahad, R. Basak, and A. Jannat, "Modeling and implementation of a mobile robotic arm for industrial tasks," *American Academic & Scholarly Research Journal*, vol. 5, no. 2, pp. 70–79, Mar. 2013. [Online]. Available: http://search.proquest.com/docview/1399138896/abstract?accountid=27795

[11] Z. M. Long, S. Q. Guo, G. J. Chen, and B. L. Yin, "Modeling and simulation for the articulated robotic arm test system of the combination drive," *Applied Mechanics and Materials*, vol. 151, Jan. 2012. [Online]. Available: http://search.proquest.com/docview/1443000029/abstract?accountid=27795#

[12] P. J. Wallin, "Robotics in the food industry: an update," *Trends in Food Science & Technology*, vol. 8, no. 6, pp. 193–198, Jun. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092422449701042X

[13] D. Meike and L. Ribickis, "Energy efficient use of robotics in the automobile industry," in *2011 15th International Conference on Advanced Robotics (ICAR)*, Jun. 2011, pp. 507–511.

[14] F. Erzincanli and J. M. Sharp, "A classification system for robotic food handling," *Food Control*, vol. 8, no. 4, pp. 191–197, Aug. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0956713597000480

[15] M. Ouerfelli, V. Kumar, and W. S. Harwin, "Methods for kinematic modeling of biological and robotic systems," *Medical Engineering & Physics*, vol. 22, no. 7, pp. 509–520, Sep. 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350453300000631

[16] L. Žlajpah, "Simulation in robotics," *Mathematics and Computers in Simulation*, vol. 79, no. 4, pp. 879–897, Dec. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378475408001183

[17] G. López-Nicolás, A. Romeo, and J. J. Guerrero, "Active learning in robotics based on simulation tools," *Computer Applications in Engineering Education*, vol. 22, no. 3, pp. 509–515, Sep. 2014. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/cae.20576/abstract

[18] b. Wenzhe Wang1, x. Shiyue Liu1, Qingbo Geng1, and Qing Fei1, "Development of PC-based simulation and control platform for a 6-DOF robotic arm," *Applied Mechanics & Materials*, no. 543-547, pp. 1397–1400, Jun. 2014. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=95257977&site=ehost-live

[19] R. Ranjan, A. Kumar, and P. Dhyani, "Modeling and simulation of robotic humanoid arm," *International Journal of Engineering Science and Technology*, vol. 4, no. 6, pp. 2616–2625, Jun. 2012. [Online]. Available: http://search.proquest.com/docview/1026647013?pq-origsite=summon

[20] E. V. Krasilnikyants, A. A. Varkov, and V. V. Tyutikov, "Robot manipulator control system," *Automation and Remote Control*, vol. 74, no. 9, pp. 1589–1598, Sep. 2013. [Online]. Available: http://link.springer.com/article/10.1134/S0005117913090154

[21] P. S. Yadav and N. Singh, "Robust control of two link rigid manipulator." [Online]. Available: http://www.ijiee.org/vol5/530-A0016.pdf

[22] Q. X. Xia, Y. Q. Yu, and Q. B. Liu, "Fuzzy control for underactuated manipulator," *Applied Mechanics and Materials*, vol. 397-400, Sep. 2013. [Online]. Available: http://search.proquest.com/docview/1442195054/abstract?accountid=27795#

[23] M. Soylemez, M. Gokasan, and O. Bogosyan, "Position control of a single-link robot-arm using a multi-loop PI controller," in *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003*, vol. 2, Jun. 2003, pp. 1001–1006 vol.2.

[24] F. Piltan, R. Bayat, F. Aghayari, and B. Boroomand, "Design error-based linear model-free evaluation performance computed torque controller," *International Journal of Robotics and Automation*, vol. 3, no. 3, pp. 151–166, 2012. [Online]. Available: http://www.researchgate.net/profile/Farzin_Piltan3/publication/263969238_Design_Error-based_Linear_Model-free_Evaluation_Performance_Computed_Torque_Controller/links/00b4953c7749e749ec000000.pdf

[25] F. Piltan, N. Sulaiman, M. H. Marhaban, A. Nowzary, and M. Tohidian, "Design of FPGA-based sliding mode controller for robot manipulator," *International Journal of Robotics and Automation (IJRA)*, vol. 2, no. 3, pp. 173–194, 2011. [Online]. Available: http://www.academia.edu/download/30900697/IJRA_V2_I3.pdf#page=52

[26] S. Z. S. Al-Khayyt, "Tuning PID controller by neural network for robot manipulator trajectory tracking," *Al-Khwarizmi Engineering Journal*, vol. 8, no. 1, pp. 19–28, 2013. [Online]. Available: http://aliraq.ws/LionImages/News/69953.pdf

[27] D. Receanu, "Modeling and simulation of the nonlinear computed torque control in simulink/MATLAB for an industrial robot," *SL: Structural Longevity*, vol. 10, no. 2, pp. 95–106, 2013. [Online]. Available: http://www.techscience.com/doi/10.3970/sl.2013.010.095.pdf

[28] A. Menon, B. Cohen, and M. Likhachev, "Motion planning for smooth pickup of moving objects," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 453–460. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6906895

[29] R. W. Brockett, "Robotic manipulators and the product of exponentials formula," in *Mathematical theory of networks and systems*. Springer, 1984, pp. 120–129. [Online]. Available: http://link.springer.com/chapter/10.1007/BFb0031048

[30] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994. [Online].

Available: http://books.google.com/books?hl=en&lr=&id=D_PqGKRo7oIC&oi=fnd&pg=PR13&dq=%22This+period+has+been+accompanied+by+a+technological%22+%22with+the+growth+in+robotics+in+the+last+two+decades%22+%22the+state+of+maturity+of+the+subject+and+the+vast+diversity+of%22+&ots=dkNlwiZ4po&sig=3dxTWr5SIygwZkoE0b_QHHzsPhc

[31] X. Xiao, Y. Li, and H. Tang, "Kinematics and interactive simulation system modeling for robot manipulators," in *Information and Automation (ICIA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1177–1182. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6720473

[32] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.

[33] P. I. Corke and others, "A computer tool for simulation and analysis: the robotics toolbox for MATLAB," in *Proc. National Conf. Australian Robot Association*, 1995, pp. 319–330. [Online]. Available: http://www.pessoal.utfpr.edu.br/winderson/arquivos/ARA95.pdf

[34] P. I. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Berlin: Springer, 2011.

[35] P. Corke, *Robotics, Vision and Control*, ser. Springer Tracts in Advanced Robotics, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 73. [Online]. Available: http://link.springer.com/10.1007/978-3-642-20144-8

[36] "http://www.imagesco.com/kits/robotic-arm.html."

[37] "RA-01c.pdf." [Online]. Available: http://www.imagesco.com/catalog/robot/RA-01C.pdf

[38] "http://www.imagesco.com/robotics/arm.html."

[39] M. D. Ardema, "Lagrange's equations," *Analytical Dynamics: Theory and Applications*, pp. 109–130, 2005. [Online]. Available: http://link.springer.com/content/pdf/10.1007/0-306-48682-2_6.pdf

[40] L. A. Soriano, J. de Jesus Rubio, S. Rodriguez, and C. Torres, "Dynamic model for an articulated manipulator," *ICIC express letters Part B Applications*, vol. 2, no. 2, pp. 415–420, 2011. [Online]. Available: http://www.researchgate.net/profile/Jose_de_Jesus_Rubio/publication/239521555_Dynamic_model_for_an_articulated_manipulator/links/0deec52583592b9151000000.pdf

[41] P. J. From, J. T. Gravdahl, and K. Y. Pettersen, *Vehicle-Manipulator Systems*, ser. Advances in Industrial Control. London: Springer London, 2014. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-5463-1

[42] F. Fahimi, *Autonomous Robots*. Boston, MA: Springer US, 2009. [Online]. Available: http://link.springer.com/10.1007/978-0-387-09538-7

[43] R. Kelly, V. Santibańẽz, and A. Loriá, *Control of robot manipulators in joint space*. London: Springer, 2005. [Online]. Available: http://site.ebrary.com/id/10229185

[44] V. Arunachalam, "Optimization using differential evolution," 2008. [Online]. Available: http://ir.lib.uwo.ca/wrrr/22/

[45] Musrrat Ali, M. Pant, and A. Abraham, "Simplex differential evolution," *Acta Polytechnica Hungarica*, vol. 6, no. 5, pp. 95–115, 2009. [Online]. Available: http://www.researchgate.net/profile/Musrrat_Ali/publication/45087902_Simplex_Differential_Evolution/links/0912f50639b26cf203000000.pdf

[46] P. Corke, "A robotics toolbox for MATLAB," *IEEE Robotics Automation Magazine*, vol. 3, no. 1, pp. 24–32, Mar. 1996.

APPENDIX A

PHYSICAL PARAMETERS

The physical parameters used are shown in table (1.1) as per the servos specifications (5.2) and estimated dimensions form the available kits.

Table 1.1: Physical Parameters

Parameter	Value	Unit
r_1	0.02837	m
r_2	0.10400	m
r_3	0.05000	m
r_4	0.04800	m
l_1	0.05800	m
l_2	0.12000	m
l_3	0.09000	m
l_4	0.09000	m
m_1	0.1844	kg
m_2	0.1172	kg
m_3	0.0335	kg
m_4	0.1160	kg
I_{x1}	2.6145×10^{-4}	$kg.m^2$
I_{y1}	2.5098×10^{-4}	$kg.m^2$
I_{z1}	4.8414×10^{-4}	$kg.m^2$
I_{x2}	0.26790×10^{-4}	$kg.m^2$
I_{y2}	15.150×10^{-4}	$kg.m^2$
I_{z2}	15.120×10^{-4}	$kg.m^2$
I_{x3}	0.06302×10^{-4}	$kg.m^2$
I_{y3}	2.1731×10^{-4}	$kg.m^2$
I_{z3}	2.2819×10^{-4}	$kg.m^2$
I_{x4}	0.3595×10^{-4}	$kg.m^2$
I_{y4}	2.8595×10^{-4}	$kg.m^2$
I_{z4}	2.7753×10^{-4}	$kg.m^2$

APPENDIX B

SERVO SPECIFICATIONS

A detailed specifications of the servos are given as follows [36]:

Detailed Servo Specifications
"Control System: +Pulse Width Control 1500usec Neutral
Required Pulse: 3-5 Volt Peak to Peak Square Wave
Operating Voltage: 4.8-6.0 Volts
Operating Temperature Range: -20 to +60 Degree C
Operating Speed (4.8V): 0.16sec/60 degrees at no load
Operating Speed (6.0V): 0.13sec/60 degrees at no load
Stall Torque (4.8V): 76.37 oz/in. (5.5kg.cm)
Stall Torque (6.0V): 94.43 oz/in. (6.8kg.cm)
Operating Angle: 45 Deg. one side pulse traveling 400usec
360 Modifiable: Yes
Direction: Clockwise/Pulse Traveling 1500 to 1900usec
Current Drain (4.8V): 8.8mA/idle and 400mA no load operating
Current Drain (6.0V): 9.1mA/idle and 500mA no load operating
Dead Band Width: 8usec
Motor Type: 3 Pole Ferrite
Potentiometer Drive: Indirect Drive
Bearing Type: Dual Ball Bearing
Gear Type: 3 Metal Gears and 1 Resin Metal Gear
Connector Wire Length: 11.81" (300mm)
Dimensions: 1.59" x 0.77"x 1.48" (40.6 x 19.8 x 37.8mm)
Weight: 1.94oz. (55.2g)
Operating Speed: 0.18 / 0.15 sec.
Output Torque: 76 / 94 oz. 5.5 / 6.8 kg.
Size: 1.6"x 0.8"x 1.5" 41 x 20 x 38mm
Weight: 1.94 oz. 55.2 g.
HS-625MG - 39.95
High Speed Metal Gear Servo."

APPENDIX C

JAVA CODE

```
// NSTIP Project RobotController Java program

// Use RXTX instead of javax
import gnu.io.CommPort;
import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;

import java.io.FileDescriptor;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public class NSTIP_RobotControl
{
public NSTIP_RobotControl()
{
```

```
super();
}

// Open port
void connect ( String portName ) throws Exception
{
CommPortIdentifier portIdentifier =  CommPortIdentifier.getPortIdentifie
(portName);
if ( portIdentifier.isCurrentlyOwned() )
{
System.out.println("Error: Port is currently in use");
}
else
{
CommPort commPort = portIdentifier.open(this.getClass().getName(),2000);

if ( commPort instanceof SerialPort )
{
SerialPort serialPort = (SerialPort) commPort;
serialPort.setSerialPortParams(19200,SerialPort.DATABITS_8,
SerialPort.

 STOPBITS_1,SerialPort.PARITY_NONE);

InputStream in = serialPort.getInputStream();
OutputStream out = serialPort.getOutputStream();
```

```
(new Thread(new SerialReader(in))).start();
(new Thread(new SerialWriter(out))).start();

}
else
{
System.out.println("Error: Only serial ports are handled by

 this example.");
}
}
}

// Reader
public static class SerialReader implements Runnable
{
InputStream in;

public SerialReader ( InputStream in )
{
this.in = in;
}

public void run ()
{
byte[] buffer = new byte[1024];
int len = -1;
```

```
try
{
while ( ( len = this.in.read(buffer)) > -1 )
{
System.out.print(new String(buffer,0,len));
}
}
catch ( IOException e )
{
e.printStackTrace();
}
}
}

// Writer
public static class SerialWriter implements Runnable
{
OutputStream out;

public SerialWriter ( OutputStream out )
{
this.out = out;
}

public void run ()
{
try
```

```
{
//int c = 0;
//while ( ( c = System.in.read()) > -1 )
//{
//     this.out.write(c);
//}

// GoTo a position
// 4D 01 FF 03 D0 00 FF 00 FF 0D
this.out.write((char) 0x4d);
this.out.write((char) 0x01);
this.out.write((char) 0xff);
this.out.write((char) 0x03);
this.out.write((char) 0xd0);
this.out.write((char) 0x00);
this.out.write((char) 0xff);
this.out.write((char) 0x00);
this.out.write((char) 0xff);
this.out.write((char) 0x0d);

try {
Thread.sleep(2000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}

// Grip
```

```
this.out.write('G');
this.out.write('\r');

try {
Thread.sleep(2000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}

// GoTo a position
// 4d 01 16 03 0a 00 f9 02 87 0d
this.out.write((char) 0x4d);
this.out.write((char) 0x01);
this.out.write((char) 0x16);
this.out.write((char) 0x03);
this.out.write((char) 0x0a);
this.out.write((char) 0x00);
this.out.write((char) 0xf9);
this.out.write((char) 0x02);
this.out.write((char) 0x87);
this.out.write((char) 0x0d);

try {
Thread.sleep(2000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}
```

```
// Grip
this.out.write('R');
this.out.write('\r');



}
catch ( IOException e )
{
e.printStackTrace();
}
}
}

public static void main ( String[] args )
{
try
{
(new NSTIP_RobotControl()).connect("COM4");
}
catch ( Exception e )
{
// TODO Auto-generated catch block
e.printStackTrace();
}
}
}
```

VITAE

- **Personal Details:**
  - Name: Amin Abdelgader Mohammed Fadlalla
  - Nationality: Sudanese
  - Date of Birth: 1^{st}, January, 1988
  - Email: *www.alhedor.h@gmail.com*
  - Permenant Address: Alhedor, East of El-Gezira, El-Gezira State, Sudan

- **Education:**
  1. Bachelor of Science, University of Khartoum, Sudan. Graduated August 2011.
  2. Master of Science, King Fahd University of Petroleum & Minerals, Saudi Arabia. Graduating May 2015.